Tree-based construction of LDPC codes having good pseudocodeword weights

Kelley, C A ; Sridhara, D ; Rosenthal, J

Abstract: We present a tree-based construction of low-density parity-check (LDPC) codes that have minimum pseudocodeword weight equal to or almost equal to the minimum distance, and perform well with iterative decoding. The construction involves enumerating a d-regular tree for a fixed number of layers and employing a connection algorithm based on permutations or mutually orthogonal Latin squares to close the tree. Methods are presented for degrees d=ps and d=ps+1, for p a prime. One class corresponds to the well-known finite-geometry and finite generalized quadrangle LDPC codes; the other codes presented are new. We also present some bounds on pseudocodeword weight for p-ary LDPC codes. Treating these codes as p-ary LDPC codes rather than binary LDPC codes improves their rates, minimum distances, and pseudocodeword weights, thereby giving a new importance to the finite-geometry LDPC codes where p>2

DOI: https://doi.org/10.1109/TIT.2007.892774

Tree-Based Construction of LDPC Codes Having Good Pseudocodeword Weights

Christine Kelley
Department of Mathematics
University of Notre Dame
Notre Dame, IN 46556, U.S.A.
email: ckelley1@nd.edu

Deepak Sridhara and Joachim Rosenthal
Institut für Mathematik
Universität Zürich
CH-8057, Zürich, Switzerland.
email: {sridhara, rosen}@math.unizh.ch

Abstract

We present a tree-based construction of LDPC codes that have minimum pseudocodeword weight equal to or almost equal to the minimum distance, and perform well with iterative decoding. The construction involves enumerating a d-regular tree for a fixed number of layers and employing a connection algorithm based on permutations or mutually orthogonal Latin squares to close the tree. Methods are presented for degrees $d = p^s$ and $d = p^s + 1$, for p a prime. One class corresponds to the well-known finite-geometry and finite generalized quadrangle LDPC codes; the other codes presented are new. We also present some bounds on pseudocodeword weight for p-ary LDPC codes. Treating these codes as p-ary LDPC codes rather than binary LDPC codes improves their rates, minimum distances, and pseudocodeword weights.

Index Terms

Low density parity check codes, pseudocodewords, iterative decoding, min-sum iterative decoding, p-ary pseudoweight.

I. Introduction

Low Density Parity Check (LDPC) codes are widely acknowledged to be good codes due to their near Shannon-limit performance when decoded iteratively. However, many structure-based constructions of LDPC codes fail to achieve this level of performance, and are often outperformed by random constructions. (Exceptions include the finite-geometry-based LDPC codes (FG-LDPC) of [9], which were later generalized in [15].) Moreover, there are discrepancies between iterative and maximum likelihood (ML) decoding performance of short to moderate blocklength LDPC codes. This behavior has recently been attributed to the presence of so-called *pseudocodewords* of the LDPC constraint graphs (or, Tanner graphs), which are valid solutions of the iterative decoder which may or may not be optimal [8]. Analogous to the role of minimum Hamming distance, $d_{\min}$, in ML-decoding, the minimum pseudocodeword weight[1], $w_{\min}$, has been shown to be a leading predictor

This work was supported in part by the NSF Grant No. CCR-ITR-02-05310. Some of the material in this paper was previously presented at ISIT 2005 (Adelaide, Australia).

[1]Note that the minimum pseudocodeword weight is specific to the LDPC graph representation of the LDPC code.

of performance in iterative decoding [19]. Furthermore, the error floor performance of iterative decoding is dominated by minimum weight pseudocodewords. Although there exist pseudocodewords with weight larger than $d_{\min}$ that have adverse affects on decoding, it has been observed that pseudocodewords with weight $w_{\min} < d_{\min}$ are especially problematic [6].

The Type I-A construction and certain cases of the Type II construction presented in this paper are designed so that the resulting codes have minimum pseudocodeword weight equal to or almost equal to the minimum distance of the code, and consequently, these problematic low-weight pseudocodewords are avoided. Some of the resulting codes have minimum distance which meets the lower tree bound originally presented in [16]. Since $w_{\min}$ shares the same lower bound [6], [7], and is upper bounded by $d_{\min}$, these constructions have $w_{\min} = d_{\min}$. It is worth noting that this property is also a characteristic of some of the FG -LDPC codes [15], and indeed, the projective-geometry-based codes of [9] arise as special cases of our Type II construction. Furthermore, the Type I-B construction presented herein yields a family of codes with a wide range of rates and blocklengths that are comparable to those obtained from finite geometries. This new family of codes has $w_{\min} = d_{\min} \geq$ tree bound in most cases.

Both min-sum and sum-product iterative decoding performance of the tree-based constructions are comparable to, if not, better than, that of random LDPC codes of similar rates and block lengths. We now present the tree bound on $w_{\min}$ derived in [7].

Definition 1.1: The tree bound of a d left (variable node) regular bipartite LDPC constraint graph with girth g is defined as

$$T(d,g) := \begin{cases} 1 + d + d(d-1) + d(d-1)^2 + \ldots + d(d-1)^{\frac{g-6}{4}}, & \frac{g}{2} \text{ odd} , \\ 1 + d + d(d-1) + \ldots + d(d-1)^{\frac{g-8}{4}} + (d-1)^{\frac{g-4}{4}}, & \frac{g}{2} \text{ even} . \end{cases} \tag{1}$$

Theorem 1.2: Let G be a bipartite LDPC constraint graph with smallest left (variable node) degree d and girth g. Then the minimum pseudocodeword weight $w_{\min}$ (for the AWGN/BSC channels) is lower bounded by

$$w_{\min} \geq T(d,g).$$

This bound is also the tree bound on the minimum distance established by Tanner in [16]. And since the set of pseudocodewords includes all codewords, we have $w_{\min} \leq d_{\min}$.

The paper is organized as follows. The Type I constructions are presented in Section 2 and properties of the resulting codes are discussed. Section 3 presents the Type II construction with two variations and the resulting codes are compared with codes arising from finite geometries and finite generalized quadrangles. In Section 4, we provide simulation results of the codes on the additive white Gaussian noise (AWGN) channel and on the p-ary symmetric channel. The paper is concluded in Section 5.

II. Tree-based Construction: Type I

In the Type I construction, first a d-regular tree of alternating variable and constraint node layers is enumerated from a root variable node (layer L_0) for ℓ layers. If ℓ is odd (respectively, even), the final layer $L_{\ell-1}$ is composed of variable nodes (respectively, constraint nodes). Call this tree T. The tree T is then reflected across an imaginary horizontal axis to yield another tree, T', and the variable and constraint nodes are reversed. That is, if layer L_i in T is composed of variable nodes, then the reflection of L_i, call it L'_i, is composed of constraint nodes in the reflected tree, T'. The union of these two trees, along with edges connecting the nodes in layers $L_{\ell-1}$ and $L'_{\ell-1}$ according to a connection algorithm that is described next, comprise the graph representing a Type I LDPC code. We now present two connection schemes that can be used in this Type I model, and discuss the resulting codes.

A. Type I-A

For $d = 3$, the Type I-A construction yields a d-regular LDPC constraint graph having $1+d+d(d-1)+\ldots+d(d-1)^{\ell-2}$ variable and constraint nodes, and girth g. The tree T has ℓ layers. To connect the nodes in $L_{\ell-1}$ to $L'_{\ell-1}$, first label the variable (resp., constraint) nodes in $L_{\ell-1}$ (resp., $L'_{\ell-1}$) when ℓ is odd (and vice versa when ℓ is even), as $v_0, v_1, \ldots, v_{2^{\ell-2}-1}$, $v_{2^{\ell-2}}, \ldots, v_{2\cdot 2^{\ell-2}-1}, v_{2\cdot 2^{\ell-2}}, \ldots, v_{3\cdot 2^{\ell-2}-1}$ (resp., $c_0, c_1, \ldots, c_{3\cdot 2^{\ell-2}-1}$). The nodes $v_0, v_1, \ldots, v_{2^{\ell-2}-1}$ form the 0^{th} class S_0, the nodes $v_{2^{\ell-2}}, \ldots, v_{2\cdot 2^{\ell-2}-1}$ form the 1^{st} class S_1, and the nodes $v_{2\cdot 2^{\ell-2}}, \ldots, v_{3\cdot 2^{\ell-2}-1}$ form the 2^{nd} class S_2; classify the constraint nodes into S'_0, S'_1, and S'_2 in a similar manner. In addition, define four permutations $\pi(\cdot), \tau(\cdot), \tau'(\cdot), \tau''(\cdot)$ of the set $\{0, 1, \ldots, 2^{\ell-2}-1\}$ and connect the nodes in $L_{\ell-1}$ to $L'_{\ell-1}$ as follows: For $j = 0, 1, \ldots, 2^{\ell-2}-1$,

1) The variable node v_j is connected to nodes $c_{\pi(j)}$ and $c_{\tau(j)+2^{\ell-2}}$.
2) The variable node $v_{j+2^{\ell-2}}$ is connected to nodes $c_{\pi(j)+2^{\ell-2}}$ and $c_{\tau'(j)+2\cdot 2^{\ell-2}}$.
3) The variable node $v_{j+2\cdot 2^{\ell-2}}$ is connected to nodes $c_{\pi(j)+2\cdot 2^{\ell-2}}$ and $c_{\tau''(j)}$.

The permutations for the cases $g = 6, 8, 10, 12$ are given in Table I. For $\ell = 3, 4, 5, 6$, these permutations yield girths $g = 6, 8, 10, 12$, respectively, – i.e., $g = 2\ell$. It is clear that the girth of these graphs is upper bounded by 2ℓ. What is interesting is that there exist permutations π, τ, τ', τ'' that achieve this upper bound when $\ell \leq 6$. However, when extending this particular construction to $\ell = 7$ layers, there are no permutations π, τ, τ', τ'' that yield a girth $g = 14$ graph. (This was verified by an exhaustive computer search and computing the girths of the resulting graphs using MAGMA [10].) The above algorithm to connect the nodes in layers $L_{\ell-1}$ and $L'_{\ell-1}$ is rather restrictive, and we need to examine other connection algorithms that may possibly yield a girth 14 bipartite graph. However, the smallest known 3-regular graph with girth 14 has 384 vertices [1]. For $\ell = 7$, the graph of the Type I-A construction has a total of 380 nodes (i.e., 190 variable nodes and 190 constraint nodes), and there are permutations π, τ, τ', and τ'', that only result in a girth 12 (bipartite) graph.

When $\ell = 3, 5$, the minimum distance of the resulting code meets the tree bound, and hence, $d_{\min} = w_{\min}$. When $\ell = 4, 6$, the minimum distance $d_{\min}$ is strictly larger than the tree bound; in fact, $d_{\min}$ is more than

Girth	$g=6$	$g=8$	$g=10$	$g=12$
π	(0)(1)	(0)(2)(1,3)	(0)(2)(4)(6)(1,5)(3,7)	(0)(4)(8)(12)(2,6)(10,14)(1,9)(3,15)(5,13)(7,11)
τ	$=\pi$	$=\pi$	(0)(2)(4)(6)(1,7)(3,5)	(0)(4)(8)(12)(2,6)(10,14)(1,13)(3,11)(5,9)(7,15)
τ'	$=\pi$	$=\pi$	$=\tau$	(0,8)(4,12)(2,14)(6,10)(1,5)(3)(7)(9,13)(11)(15)
τ''	$=\pi$	(0,2)(1)(3)	(0,4)(2,6)(1,3)(5,7)	(0,2,4,6)(8,10,12,14)(1,15,5,11)(3,9,7,13)

TABLE I

PERMUTATIONS FOR TYPE I-A CONSTRUCTION.

the tree-bound by 2. However, $w_{\min} = d_{\min}$ for $\ell = 4, 6$ as well. Figure 3 illustrates the general construction procedure and Figure 4 shows a 3-regular girth 10 Type I-A LDPC constraint graph.

Remark 2.1: The Type I-A LDPC codes have $d_{\min} = w_{\min} = T(d, 2\ell)$, for $\ell = 3, 5$, and $d_{\min} = w_{\min} = 2 + T(d, 2\ell)$, for $\ell = 4, 6$.

B. Mutually orthogonal Latin squares

Let $\mathbb{F} := GF(q)$ be a finite field of order q and let $\mathbb{F}^*$ denote the corresponding multiplicative group –i.e., $\mathbb{F}^* := \mathbb{F}\backslash\{0\}$. For every $a \in \mathbb{F}^*$, we define a $q \times q$ array having entries in $\mathbb{F}$ by the following linear map

$$M_a : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$$

$$(x, y) \mapsto x + a \cdot y$$

where '$+$' and '$\cdot$' are the corresponding field operations. The above set of maps define $q-1$ mutually orthogonal Latin squares (MOLS) [pg. 182 – 199, [17]]. By introducing an array M_0 defined in the following manner

$$M_0 : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$$

$$(x, y) \mapsto x$$

we obtain an additional array which is orthogonal to the above family of $q-1$ MOLS. However, note that M_0 is not a Latin square. We use this set of q arrays in the subsequent tree-based constructions.

C. Type I-B

For $d = p^s$, a prime power, the Type I-B construction yields a d-regular LDPC constraint graph having $1 + d + d(d-1)$ variable and constraint nodes, and girth at least 6. The tree T has 3 layers L_0, L_1, and L_2. The tree is reflected to yield another tree T' and the variable and constraint nodes in T' are interchanged. Let α be a primitive element in the field $GF(p^s)$. (Note that $GF(p^s)$ is the set $\{0, 1, \alpha, \alpha^2, \ldots, \alpha^{p^s-2}\}$.) The layer L_1 (resp., L_1') contains p^s constraint nodes labeled $(0)_c, (1)_c, (\alpha)_c, \ldots, (\alpha^{p^s-2})_c$ (resp., variable nodes labeled $(0)', (1)', (\alpha)', \ldots, (\alpha^{p^s-2})'$). The layer L_2 (resp., L_2') is composed of p^s sets $\{S_i\}_{i=0,1,\alpha,\alpha^2,\ldots,\alpha^{p^s-2}}$

of $p^s - 1$ variable (resp., constraint) nodes in each set. Note that we index the sets by an element of the field $GF(p^s)$. Each set S_i corresponds to the children of one of the branches of the root node. (The '$\prime$' in the labeling refers to nodes in the tree T' and the subscript 'c' refers to constraint nodes.) Let S_i (resp., S'_i) contain the variable nodes $(i,1),(i,\alpha),\ldots,(i,\alpha^{p^s-2})$ (resp., constraint nodes $(i,1)'_c,(i,\alpha)'_c,\ldots,(i,\alpha^{p^s-2})'_c$). To use MOLS of order p^s in the connection algorithm, an imaginary node, variable node $(i,0)$ (resp., constraint node $(i,0)'_c$) is temporarily introduced into each set S_i (resp, S'_i). The connection algorithm proceeds as follows:

1) For $i = 0,1,\alpha,\ldots,\alpha^{p^s-2}$ and $j = 0,1,\alpha,\ldots,\alpha^{p^s-2}$, connect the variable node (i,j) in layer L_2 to the constraint nodes

$$(0, j+i\cdot 0)'_c,\ (1, j+i\cdot 1)'_c,\ \ldots,\ (\alpha^{p^s-2}, j+i\cdot\alpha^{p^s-2})'_c$$

in layer L'_2. (Observe that in these connections, every variable node in the set S_i is mapped to exactly one constraint node in each set S'_k, for $k = 0,1,\alpha,\ldots,\alpha^{p^s-2}$, using the array M_i defined in Section 2.B.)

2) Delete all imaginary nodes $\{(i,0),(i,0)'_c\}_{i=0,1,\ldots,\alpha^{p^s-2}}$ and the edges incident on them.

3) For $i = 1,\ldots,\alpha^{p^s-2}$, delete the edge connecting variable node $(0,i)$ to constraint node $(0,i)'_c$.

The resulting d-regular constraint graph represents the Type I-B LDPC code. Figure 5 illustrates the construction procedure and Figure 6 provides a specific example of a Type I-B LDPC constraint graph with $d = 4 = 2^2$; the arrays used for constructing this graph are

$$M_0 := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ \alpha & \alpha & \alpha & \alpha \\ \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2 \end{bmatrix},\ M_1 := \begin{bmatrix} 0 & 1 & \alpha & \alpha^2 \\ 1 & 0 & \alpha^2 & \alpha \\ \alpha & \alpha^2 & 0 & 1 \\ \alpha^2 & \alpha & 1 & 0 \end{bmatrix},\ M_\alpha := \begin{bmatrix} 0 & \alpha & \alpha^2 & 1 \\ 1 & \alpha^2 & \alpha & 0 \\ \alpha & 0 & 1 & \alpha^2 \\ \alpha^2 & 1 & 0 & \alpha \end{bmatrix},\ M_{\alpha^2} := \begin{bmatrix} 0 & \alpha^2 & 1 & \alpha \\ 1 & \alpha & 0 & \alpha^2 \\ \alpha & 1 & \alpha^2 & 0 \\ \alpha^2 & 0 & \alpha & 1 \end{bmatrix}.$$

The Type I-B algorithm yields LDPC codes having a wide range of rates and blocklengths that are comparable to, but different from, the two-dimensional LDPC codes from finite Euclidean geometries [9], [15]. The Type I-B LDPC codes are p^s-regular with girth at least six, blocklength $N = p^{2s}+1$, and distance $d_{\min} \geq p^s + 1$. For degrees of the form $d = 2^s$, the resulting binary Type I-B LDPC codes have very good rates, above 0.5, and perform well with iterative decoding.

Theorem 2.2: The Type I-B LDPC constraint graphs have a girth of at least six.

Proof: We need to show that there are no 4-cycles in the graph. By construction, it is clear that there are no 4-cycles that involve the nodes in layers L_0, L'_0, L_1, and L'_1. This is because no two nodes, say, variable nodes (i,j) and (i,k) in a particular class S_i are connected to the same node $(s,t)'_c$ in some class S'_s; otherwise, it would mean that $t = j + i\cdot s = k + i\cdot s$. But this is only true for $j = k$. Therefore, suppose there is a 4-cycle in the graph, then let us assume that variable nodes (i,j) and (s,t), for $s \neq i$, are each connected to constraint nodes $(a,b)'_c$ and $(e,f)'_c$. By construction, this means that $b = j + i\cdot a = t + s\cdot a$ and $f = j + i\cdot e = t + s\cdot e$. However then $j - t = (s-i)\cdot a = (s-a)\cdot e$, thereby implying that $a = e$. When $a = e$, we also have $b = j + i\cdot a = j + i\cdot e = f$. Thus, $(a,b)'_c = (e,f)'_c$. Therefore, there are no 4-cycles in the Type I-B LDPC graphs. ■

Theorem 2.3: The Type I-B LDPC constraint graphs with degree $d = p^s$ and girth $g \geq 6$ have $2(p^s - 1) \geq d_{\min} \geq w_{\min} \geq T(p^s, 6) = 1 + p^s$ for $p > 2$ and, $2(p^s) + 1 \geq d_{\min} \geq w_{\min} \geq T(p^s, 6) = 1 + p^s$ for $p = 2$.

Proof: When p is an odd prime, the assertion follows immediately. Consider the following active variable nodes to be part of a codeword: variable nodes $(0, 1), (0, \alpha), \ldots, (0, \alpha^{p^s-2})$ in S_0, and all but the first variable node in the middle layer L'_1 of the reflected tree T': i.e., variable nodes $(1)', (\alpha)', (\alpha^2)', \ldots, (\alpha^{p^s-2})'$ in L'_1. Clearly all the constraints in L'_2 are either connected to none or exactly two of these active variable nodes. The root node in T' is connected to $p^s - 1$ (an even number) active variable nodes and the first constraint node in L_1 of T is also connected to $p^s - 1$ active variable nodes. Hence, these $2(p^s - 1)$ active variable nodes form a codeword. This fact along with Theorems 1.2 and 2.2 prove that $2(p^s-1) \geq d_{\min} \geq w_{\min} \geq T(p^s, 6) = 1+p^s$.

When $p = 2$, consider the following active variable nodes to be part of a codeword: the root node, variable nodes $(0, 1), (0, \alpha), \ldots, (0, \alpha^{p^s-2})$ in S_0, variable node (α^i, α^i) from S_{α^i}, for $i = 0, 1, 2, \ldots, p^s - 2$, and the first two variable nodes in the middle layer of T' (i.e., variable nodes $(0)'$,$(1)'$). Since $p = 2$, $p^s - 1$ is odd. We need to show that all the constraints are satisfied for this choice of active variable nodes. Each constraint node in the layer L_1 of T has an even number of active variable node neighbors: $(0)_c$ has p^s active neighbors, and $(i)_c$, for $i = 1, \alpha, \ldots, \alpha^{p^s-2}$, has two, the root node and variable node (i, i). It remains to check the constraint nodes in T'.

In order to examine the constraints in layer L'_2 of T', observe that the variable node $(0, \alpha^j)$, for $j = 1, 2, \ldots, p^s - 2$, is connected to constraint nodes

$$(1, \alpha^j)'_c, (\alpha, \alpha^j)'_c, \ldots, (\alpha^{p^s-2}, \alpha^j)'_c,$$

and the variable node (α^i, α^i), for $i = 0, 1, 2, \ldots, p^s - 2$, is connected to constraint nodes

$$(0, \alpha^i)'_c, (1, \alpha^i + \alpha^i)'_c, (\alpha, \alpha^i + \alpha^{i+1})'_c, \ldots, (\alpha^k, \alpha^{i+k})'_c, \ldots, (\alpha^{p^s-2}, \alpha^{i+p^s-2})'_c$$

Therefore, the constraint nodes $(0, \alpha^j)'_c$, for $j = 1, 2, \ldots, p^s - 2$, in S'_0 of L'_2 are connected to exactly one active variable node from layer L_2, i.e., variable node (α^j, α^j); the other active variable node neighbor is variable node $(0)'$ in the middle layer of T'. Thus, all constraints in S'_0 are satisfied.

The constraint nodes $(1, \alpha^j)'_c$, for $j = 1, 2, \ldots, p^s - 2$, in S'_1 are each connected to exactly one active variable node from L_2, i.e., variable node $(0, \alpha^j)$ from S_0. This is because, all the remaining active variable nodes in L_2, (α^i, α^i) connect to the imaginary node $(1, 0)'_c$ in S'_1 (since $(1, \alpha^i + \alpha^i)'_c = (1, 0)'_c$ when the characteristic of the field $GF(p^s)$ is $p = 2$). Thus, all constraint nodes in S'_1 have two active variable node neighbors, the other active neighbor being the variable node $(1)'$ in the middle layer of T'.

Now, let us consider the constraint nodes in S'_{α^k}, for $k = 1, 2, \ldots, p^s - 2$. The active variable nodes (α^i, α^i), for $i = 0, 1, 2, \ldots, p^s - 2$, are connected to the following constraint nodes

$$(\alpha^k, 1 + \alpha^k)'_c, (\alpha^k, \alpha + \alpha^{k+1})'_c, \ldots, (\alpha^k, \alpha^{p^s-2} + \alpha^{p^s-2+k})'_c,$$

respectively, in class S'_{α^k}. Since $\alpha^r + \alpha^{k+r} \neq \alpha^t + \alpha^{k+t}$ for $r \neq t$, the variable nodes (α^i, α^i), for $i = 0, 1, 2, \ldots, p^s - 2$, connect to distinct nodes in S'_{α^k}. Hence, each constraint node in S'_{α^k} has exactly two active

variable node neighbors – one from S_0 and the other from the set $\{(\alpha^i, \alpha^i) |\ i = 0, 1, 2, \ldots, p^s - 2\}$.

Last, we note that the root (constraint) node in T' is connected to two active variable nodes, $(0)'$ and $(1)'$. The total number of active variable nodes is $1 + (p^s - 1) + (p^s - 1) + 2 = 2p^s + 1$. This proves that the set of $2p^s + 1$ active variable nodes forms a codeword, thereby proving the desired bound. ■

When $p > 2$, the upper bound $2(p^s - 1)$ on minimum distance $d_{\min}$ (and possibly also $w_{\min}$) was met among all the cases of the Type I-B construction we examined. We conjecture that in fact $d_{\min} = 2(p^s - 1)$ for the Type I-B LDPC codes of degree $d = p^s$ when $p > 2$. Since $w_{\min}$ is lower bounded by $1 + p^s$, we have that $w_{\min}$ is close, if not equal, to $d_{\min}$.

D. p-ary LDPC codes

Let H be a parity check matrix representing a p-ary LDPC code $\mathcal{C}$. The corresponding LDPC constraint graph G that represents H is an incidence graph of the parity check matrix as in the binary case. However, each edge of G is now assigned a weight which is the value of the corresponding non-zero entry in H. (In [4], [3], LDPC codes over $GF(q)$ are considered for transmission over binary modulated channels, whereas in [14], LDPC codes over integer rings are considered for higher-order modulation signal sets.)

For convenience, we consider the special case wherein each of these edge weights are equal to one. This is the case when the parity check matrix has only zeros and ones. Furthermore, whenever the LDPC graphs have edge weights of unity for all the edges, we refer to such a graph as a binary LDPC constraint graph representing a p-ary LDPC code $\mathcal{C}$.

We first show that if the LDPC graph corresponding to H is d-left (variable-node) regular, then the same tree bound of Theorem 1.2 holds. That is,

Lemma 2.4: If G is a d-left regular bipartite LDPC constraint graph with girth g and represents a p-ary LDPC code $\mathcal{C}$. Then, the minimum distance of the p-ary LDPC code $\mathcal{C}$ is lower bounded as

$$d_{\min} \geq T(d, g).$$

Proof: The proof is essentially the same as in the binary case. Enumerate the graph as a tree starting at an arbitrary variable node. Furthermore, assume that a codeword in $\mathcal{C}$ contains the root node in its support. The root variable node (at layer L_0 of the tree) connects to d constraint nodes in the next layer (layer L_1) of the tree. These constraint nodes are each connected to some sets of variable nodes in layer L_2, and so on. Since the graph has girth g, the nodes enumerated up to layer $L_{\frac{g-6}{2}}$ when $\frac{g}{2}$ is odd (respectively, $L_{\frac{g-4}{2}}$ when $\frac{g}{2}$ is even) are all distinct. Since the root node belongs to a codeword, say $\mathbf{c}$, it assumes a non-zero value in $\mathbf{c}$. Since the constraints must be satisfied at the nodes in layer L_1, at least one node in Layer L_2 for each constraint node in L_1 must assume a non-zero value in $\mathbf{c}$. (This is under the assumption that an edge weight times a (non-zero) value, assigned to the corresponding variable node, is not zero in the code alphabet. Since we have chosen the edge weights to be unity, such a case will not arise here. But also more generally, such

cases will not arise when the alphabet and the arithmetic operations are that of a finite field. However, when working over other structures, such as finite integer rings and more general groups, such cases could arise.)

Under the above assumption, that there are at least d variable nodes (i.e., at least one for each node in layer L_1) in layer L_2 that are non-zero in $\mathbf{c}$. Continuing this argument, it is easy to see that the number of non-zero components in $\mathbf{c}$ is at least $1 + d + d(d-1) + \cdots + d(d-1)^{\frac{g-6}{4}}$ when $\frac{g}{2}$ is odd, and $1 + d + d(d-1) + \cdots + d(d-1)^{\frac{g-8}{4}} + (d-1)^{\frac{g-4}{4}}$ when $\frac{g}{2}$ is even. Thus, the desired lower bound holds. ■

We note here that in general this lower bound is not met and typically p-ary LDPC codes that have the above graph representation have minimum distances larger than the above lower bound.

Recall from [8], [6] that a pseudocodeword of an LDPC constraint graph G is a valid codeword in some finite cover of G. To define a pseudocodeword for a p-ary LDPC code, we will restrict the discussion to LDPC constraint graphs that have edge weights of unity among all their edges – in other words, binary LDPC constraint graphs that represent p-ary LDPC codes. A finite cover of a graph is defined in a natural way as in [8] wherein all edges in the finite cover also have an edge weight of unity. For the rest of this section, let G be a LDPC constraint graph of a p-ary LDPC code $\mathcal{C}$ of block length n, and let the weights on every edge of G be unity. We define a pseudocodeword F of G as a $p \times n$ matrix of the form

$$F = \begin{bmatrix} f_{0,0} & f_{1,0} & f_{2,0} & \cdots & f_{n-1,0} \\ f_{0,1} & f_{1,1} & f_{2,1} & \cdots & f_{n-1,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{0,p-1} & f_{1,p-1} & f_{2,p-1} & \cdots & f_{n-1,p-1} \end{bmatrix},$$

where the pseudocodeword F forms a valid codeword $\hat{\mathbf{c}}$ in a finite cover $\hat{G}$ of G and $f_{i,j}$ is the fraction of variable nodes in the i^{th} variable cloud, for $0 \leq i \leq n-1$, of $\hat{G}$ that have the assignment (or, value) equal to j, for $0 \leq j \leq p-1$, in $\hat{\mathbf{c}}$.

A p-ary symmetric channel is shown in Figure 11. The input and the output of the channel are random variables belonging to a p-ary alphabet that can be denoted as $\{0, 1, 2, \ldots, p-1\}$. An error occurs with probability ϵ, which is parameterized by the channel, and in the case of an error, it is equally probable for an input symbol to be altered to any one of the remaining symbols.

Following the definition of pseudoweight for the binary symmetric channel [5], we provide the following definition for the weight of a pseudocodeword on the p-ary symmetric channel. For a pseudocodeword F, let F' be the sub-matrix obtained by removing the first row in F. (Note that the first row in F contains the entries $f_{0,0}, f_{1,0}, f_{2,0}, \ldots, f_{n-1,0}$.) Then the weight of a pseudocodeword F on the p-ary symmetric channel is defined as follows.

Definition 2.5: Let e be a number such that the sum of the e largest components in the matrix F', say, $f_{i_1,j_1}, f_{i_2,j_2}, \ldots, f_{i_e,j_e}$, exceeds $\sum_{i \neq i_1,i_2,\ldots,i_e}(1 - f_{i,0})$. Then the weight of F on the p-ary symmetric channel

is defined as

$$w_{PSC}(F) = \begin{cases} 2e, & \text{if } f_{i_1,j_1} + \cdots + f_{i_e,j_e} = \sum_{i \neq i_1,i_2,\ldots,i_e}(1 - f_{i,0}), \\ 2e-1, & \text{if } f_{i_1,j_1} + \cdots + f_{i_e,j_e} > \sum_{i \neq i_1,i_2,\ldots,i_e}(1 - f_{i,0}). \end{cases}$$

Note that in the above definition, none of the j_k's, for $k = 1, 2, \ldots, e$, are equal to zero, and all the i_k's, for $k = 1, 2, \ldots, e$, are distinct. That is, we choose at most one component in every column of F' when picking the e largest components. (See the appendix for an explanation on the above definition of "weight".)

Observe that for a codeword, the above weight definition reduces to the Hamming weight. If F represents a codeword $\mathbf{c}$, then exactly $w = wt_H(\mathbf{c})$, the Hamming weight of $\mathbf{c}$, columns in F' contain the entry 1 in some row, and the remaining entries in F' are zero. Furthermore, the matrix F has the entry 0 in the first row of these w columns and has the entry 1 in the first row of the remaining columns. Therefore, from the weight definition of F, $e = \frac{w}{2}$ and the weight of F is $2e = w$.

We define the p-ary minimum pseudocodeword weight of G (or, minimum pseudoweight) as in the binary case, i.e., as the minimum weight of a pseudocodeword among all finite covers of G, and denote this as $w_{\min}(G)$ or $w_{\min}$ when it is clear that we are referring to the graph G.

Lemma 2.6: Let G be a d-left regular bipartite graph with girth g that represents a p-ary LDPC code $\mathcal{C}$. Then the minimum pseudocodeword weight $w_{\min}$ on the p-ary symmetric channel is lower bounded as

$$w_{\min} \geq T(d,g) = \begin{cases} (1 + d + d(d-1) + d(d-1)^2 + \ldots + d(d-1)^{\frac{g-6}{4}}), & \frac{g}{2} \textit{ odd}, \\ (1 + d + d(d-1) + \ldots + d(d-1)^{\frac{g-8}{4}} + (d-1)^{\frac{g-4}{4}}), & \frac{g}{2} \textit{ even}. \end{cases}$$

The proof of this result is moved to the appendix. We note that, in general, this bound is rather loose. (The inequality in (3), in the proof of Lemma 2.6, is typically not tight.) Moreover, we expect that p-ary LDPC codes to have larger minimum pseudocodeword weights than *corresponding binary LDPC codes*. By *corresponding binary LDPC codes*, we mean the codes obtained by interpreting the given LDPC constraint graph as one representing a binary LDPC code.

E. p-ary Type I-B LDPC codes

Theorem 2.7: For degree $d = p^s$, the resulting Type I-B LDPC constraint graphs of girth g that represent p-ary LDPC codes have minimum distance and minimum pseudocodeword weight $2p^s + 1 \geq d_{\min}, w_{\min} \geq T(d,g)$.

Proof: Consider as active variable nodes the root node, all the variable nodes in S_0, the variable nodes (α^i, α^i), for $i = 0, 1, 2, \ldots, p^s - 2$, the first variable node $(0)'$ in the middle layer of T' and one other variable node $(y)'$, that we will ascertain later, in the middle layer of T'.

Since the code is p-ary (and $p > 2$), assign the value 1 to the root variable node and to all the active variable nodes in S_0. Assign the value $p-1$ to the remaining active variable nodes in L_2, (i.e., nodes (α^i, α^i), $i = 0, 1, 2, \ldots, p^s - 2$). Assign the value 1 for the variable node $(0)'$ in the middle layer of T' and assign the value $p-1$ for the variable node $(y)'$ in the middle layer of T'. We choose y in the following manner:

The variable nodes (α^i, α^i), for $i = 0, 1, 2, \ldots, p^s - 2$, are connected to the following constraint nodes

$$(\alpha^k, 1 + \alpha^k)'_c, (\alpha^k, \alpha + \alpha^{k+1})'_c, \ldots, (\alpha^k, \alpha^j + \alpha^{k+j})'_c, \ldots, (\alpha^k, \alpha^{p^s-2} + \alpha^{k+p^s-2})'_c,$$

respectively, in class S_{α^k}. Either, the above set of constraint nodes are all distinct, or they are all equal to $(\alpha^k, 0)'_c$. This is because, $\alpha^r + \alpha^{r+k} = \alpha^t + \alpha^{t+k}$ if and only if either, $r = t$ or $1 + \alpha^k = 0$. So there is only one $k \in \{0, 1, 2, \ldots, p^s - 2\}$, for which $1 + \alpha^k = 0$, and for that value of k, we set $y = \alpha^k$.

From the proof of Theorem 2.3 and the above assignment, it is easily verified that each constraint node has value zero when the sum of the incoming active nodes is take modulo p. Thus, the set of $2p^s + 1$ active variable nodes forms a codeword, and therefore $d_{\min} \leq 2p^s + 1$. Hence, from Lemmas 2.4 and 2.6, we have $T(d, 6) \leq w_{\min} \leq d_{\min} \leq 2p^s + 1$. ■

It is also observed that if the codes resulting from the Type I-B construction are treated as p-ary codes rather than binary codes when the corresponding degree in the LDPC graph is $d = p^s$, then the rates obtained are much superior; we also believe that the minimum pseudocodeword weights (on the p-ary symmetric channel) are much closer to the minimum distances for these p-ary LDPC codes.

III. Tree-based Construction: Type II

In the Type II construction, first a d-regular tree T of alternating variable and constraint node layers is enumerated from a root variable node (layer L_0) for ℓ layers $L_0, L_1, \ldots, L_{\ell-1}$, as in Type I. The tree T is not reflected; rather, a single layer of $(d-1)^{\ell-1}$ nodes is added to form layer L_ℓ. If ℓ is odd (resp., even), this layer is composed of constraint (resp., variable) nodes. The union of T and L_ℓ, along with edges connecting the nodes in layers $L_{\ell-1}$ and L_ℓ according to a connection algorithm that is described next, comprise the graph representing a Type II LDPC code. We present the connection scheme that is used for this Type II model, and discuss the resulting codes. First, we state this rather simple observation without proof: *The girth g of a Type II LDPC graph for ℓ layers is at most 2ℓ.*

The connection algorithm for $\ell = 3$ and $\ell = 4$, wherein this upper bound on girth is in fact achieved, is as follows.

A. $\ell = 3$

For $d = p^s + 1$, where p is prime and s a positive integer, a d-regular tree is enumerated from a root (variable) node for $\ell = 3$ layers L_0, L_1, L_2. Let α be a primitive element in the field $GF(p^s)$. The d constraint nodes in L_1 are labeled $(x)_c, (0)_c, (1)_c, (\alpha)_c, \ldots, (\alpha^{p^s-2})_c$ to represent the d branches stemming from the root node. Note that the first constraint node is denoted as $(x)_c$ and the remaining constraint nodes are indexed by the elements of the field $GF(p^s)$. The $d(d-1)$ variable nodes in the third layer L_2 are labeled as follows: the variable nodes descending from constraint node $(x)_c$ form the class S_x and are labeled $(x, 0), (x, 1), \ldots, (x, \alpha^{p^s-2})$, and the variable nodes descending from constraint node $(i)_c$, for $i = 0, 1, \alpha, \ldots, \alpha^{p^s-2}$, form the class S_i and are labeled $(i, 0), (i, 1), \ldots, (i, \alpha^{p^s-2})$.

A final layer $L_\ell = L_3$ of $(d-1)^{\ell-1} = p^{2s}$ constraint nodes is added. The p^{2s} constraint nodes in L_3 are labeled $(0,0)'_c, (0,1)'_c, \ldots, (0,\alpha^{p^s-2})'_c$, $(1,0)'_c$, $(1,1)'_c$, $\ldots$, $(1,\alpha^{p^s-2})'_c$, $\ldots$, $(\alpha^{p^s-2},0)'_c$, $(\alpha^{p^s-2},1)'_c$, $\ldots$, $(\alpha^{p^s-2},\alpha^{p^s-2})'_c$. (Note that the '$'$' in the labeling refers to nodes in that are not in the tree T and the subscript 'c' refers to constraint nodes.)

1) By this labeling, the constraint nodes in L_3 are grouped into $d-1 = p^s$ classes of $d-1 = p^s$ nodes in each class. Similarly, the variable nodes in L_2 are grouped into $d = p^s+1$ classes of $d-1 = p^s$ nodes in each class. (That is, the i^{th} class of constraint nodes is $S'_i = \{(i,0)'_c, (i,1)'_c, \ldots, (i,\alpha^{p^s-2})'_c\}$.)
2) The variable nodes descending from constraint node $(x)_c$ are connected to the constraint nodes in L_3 as follows. Connect the variable node (x,i), for $i = 0,1,\ldots,\alpha^{p^s-2}$, to the constraint nodes
$$(i,0)'_c,\ (i,1)'_c,\ \ldots,\ (i,\alpha^{p^s-2})'_c.$$
3) The remaining variable nodes in layer L_2 are connected to the nodes in L_3 as follows: Connect the variable node (i,j), for $i = 0,1,\alpha,\ldots,\alpha^{p^s-2}$, $j = 0,1,\ldots,\alpha^{p^s-2}$, to the constraint nodes
$$(0,j+i\cdot 0)'_c,\ (1,j+i\cdot 1)'_c,\ (\alpha,j+i\cdot\alpha)'_c,\ \ldots,\ (\alpha^{p^s-2},j+i\cdot\alpha^{p^s-2})'_c.$$
Observe that in these connections, each variable node (i,j) is connected to exactly one constraint node within each class, using the array M_i defined in Section 2.B.

Figure 7 illustrates the construction procedure and Figure 8 provides an example of a Type II LDPC constraint graph with degree $d = 4 = 3+1$ and girth $g = 6$; the arrays used for constructing this example are[2]
$$M_0 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix},\ M_1 = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix},\ M_2 = \begin{bmatrix} 0 & 2 & 1 \\ 1 & 0 & 2 \\ 2 & 1 & 0 \end{bmatrix}.$$

The ratio of minimum distance to blocklength of the resulting codes is at least $\frac{2+p^s}{1+p^s+p^{2s}}$, and the girth is six. For degrees d of the form $d = 2^s+1$, the tree bound of Theorem 1.2 on minimum distance and minimum pseudocodeword weight [16], [7] is met, i.e., $d_{\min} = w_{\min} = 2+2^s$, for the Type II, $\ell = 3$, LDPC codes. For $p > 2$, the resulting binary LDPC codes are repetition codes of the form $[n,1,n]$, i.e., $d_{\min} = n = 1+p^s+p^{2s}$. However, if we interpret the Type II $\ell = 3$ graphs, that have degree $d = p^s+1$, as the LDPC constraint graph of a p-ary LDPC code, then the rates of the resulting codes are very good and the minimum distances come close to (but are not equal to) the tree bound in Lemma 2.4. (See also [13].) We also believe that the minimum pseudocodeword weights (on the p-ary symmetric channel) are much closer to the minimum distances for these p-ary LDPC codes.

B. Relation to finite geometry codes

The codes that result from this $\ell = 3$ construction correspond to the two-dimensional projective-geometry-based LDPC (PG LDPC) codes of [15]. We state the equivalence of the tree construction and the finite projective geometry based LDPC codes in the following.

[2]Note that in this example, $GF(3) = \{0,1,2\}$, '2' being the primitive element of the field.

Theorem 3.1: The LDPC constraint graph obtained from the Type II $\ell = 3$ tree construction for degree $d = p^s + 1$ is equivalent to the incidence graph of the finite projective plane over the field $GF(p^s)$.

It has been proved by Bose [2] that a finite projective plane (in other words, a two dimensional finite projective geometry) of order m exists if and only if a complete family of orthogonal $m \times m$ Latin squares exists. The proof of this result, as presented in [12], gives a constructive algorithm to design a finite projective plane of order m from a complete family of $m \times m$ mutually orthogonal Latin squares (MOLS). It is well known that a complete family of mutually orthogonal Latin squares exists when $m = p^s$, a power of a prime, and we have described one such family in Section 2. Hence, the constructive algorithm in [12] generates the incidence graph of the projective plane $PG(2, p^s)$ from the set of $p^s - 1$ MOLS of order p^s. The only remaining step is to verify that the incidence matrix of points over lines of this projective plane is the same as the parity check matrix of variable nodes over constraint nodes of the tree-based LDPC constraint graph of the tree construction. This step is easy to verify as the constructive algorithm in [12] is analogous to the tree construction presented in this paper.

The Type II $\ell = 3$ graphs therefore correspond to the two-dimensional projective-geometry-based LDPC codes of [9]. With a little modification of the Type II construction, we can also obtain the two-dimensional Euclidean-geometry-based LDPC codes of [9], [15]. Since a two-dimensional Euclidean geometry may be obtained by deleting certain points and line(s) of a two-dimensional projective geometry, the graph of a two-dimensional EG-LDPC code [15] may be obtained by performing the following operations on the Type II, $\ell = 3$, graph:

1) In the tree T, the root node along with its neighbors, i.e., the constraint nodes in layer L_1, are deleted.
2) Consequently, the edges from the constraint nodes $(x)_c, (0)_c, (1)_c, \ldots, (\alpha^{p^s-2})_c$ to layer L_2 are also deleted.
3) At this stage, the remaining variable nodes have degree p^s, and the remaining constraint nodes have degree $p^s + 1$. Now, a constraint node from layer L_3 is chosen, say, constraint node $(0, 0)'_c$. This node and its neighboring variable nodes and the edges incident on them are deleted. Doing so removes exactly one variable node from each class of L_2, and the degrees of the remaining constraint nodes in L_3 are lessened by one. Thus, the resulting graph is now p^s-regular with a girth of six, has $p^{2s} - 1$ constraint and variable nodes, and corresponds to the two-dimensional Euclidean-geometry-based LDPC code $EG(2, p^s)$ of [15].

Theorem 3.2: The Type II $\ell = 3$ LDPC constraint graphs have girth $g = 6$ and diameter $\delta = 3$.

Proof: We need to show is that there are no 4-cycles in the graph. As in the proof of Theorem 2.2, by construction, there are no 4-cycles that involve the nodes in layers L_0 and L_1. This is because, first, no two variable nodes in the first class $S_x = \{(x, 0), (x, 1), \ldots, (x, \alpha^{p^s-2})\}$ are connected to the same constraint node. Next, if two variable nodes, say, (i, j) and (i, k) in the i^{th} class S_i, for some $i \neq x$, are connected to a constraint node $(s, t)'_c$, then it would mean that $t = j + i \cdot s = k + i \cdot s$. But this is only true for $j = k$. Hence,

there is no 4-cycle of the form $(i)_c \to (i,j) \to (s,t)'_c \to (i,k) \to (i)$. Therefore, suppose there is a 4-cycle in the graph, then let us consider two cases as follows. Case 1) Assume that variable nodes (i,j) and (s,t), for $i \neq s$ and $i \neq x \neq s$, are each connected to constraint nodes $(a,b)'_c$ and $(e,f)'_c$. By construction, this means that $b = j + i \cdot a = t + s \cdot a$ and $f = j + i \cdot e = t + s \cdot e$. This implies that $j - t = (s-i) \cdot a = (s-i) \cdot e$, thereby implying that $a = e$. Consequently, we also have $b = j + i \cdot a = j + i \cdot e = f$. Thus, $(a,b)'_c = (e,f)'_c$. Case 2) Assume that two variable nodes, one in S_x, say, (x,j), and the other in S_i, (for $i \neq x$), say, (i,k), are connected to constraint nodes $(a,b)'_c$ and $(e,f)'_c$. Then this would mean that $a = e = j$. But since (i,k) connects to exactly one constraint node whose first index is j, this case is not possible. Thus, there are no 4-cycles in the Type II-$\ell = 3$ LDPC graphs.

To show that the girth is exactly six, we see that the following nodes form a six-cycle in the graph: the root-node, the first two constraint nodes $(x)_c$ and $(1)_c$ in layer L_1, variable nodes $(x,0)$ and $(0,0)$ in layer L_2, and the constraint node $(0,0)'_c$ in layer L_3.

To prove the diameter, we first observe that the root node is at distance of at most three from any other node. Similarly, it is also clear that the nodes in layer L_1 are at a distance of at most three from any other node. Therefore, it is only necessary to show that any two nodes in layer L_2 are at most distance two apart and similarly show that any two nodes in L_3 are at most distance two apart. Consider two nodes (i,j) and (s,t) in L_2. If $s = i$, then clearly, there is a path of length two via the parent node $(i)_c$. If $s \neq i$ and $s \neq x \neq i$, then by the property of a complete family of orthogonal Latin squares there is a node $(a,b)'_c$ in L_3 such that $b = j + i \cdot a = t + s \cdot a$. This implies that (i,j) and (s,t) are connected by a distance two path via $(a,b)'_c$. We can similarly show that if $s \neq i$ and $i = x$, then the node $(j, t + s \cdot j)'_c$ in L_3 connects to both (x,j) and (s,t). A similar argument shows that any two nodes in L_3 are distance two apart. This completes the proof. ■

Theorem 3.3: For degrees $d = 2^s + 1$, the resulting Type II $\ell = 3$ LDPC constraint graphs have $d_{\min} = w_{\min} = T(d,6) = 2 + 2^s$. For degrees $d = p^s + 1$, $p > 2$, when the resulting Type II $\ell = 3$ LDPC constraint graphs represent p-ary linear codes, the corresponding minimum distance and minimum pseudocodeword weight satisfy

$$T(p^s + 1, 6) \leq w_{\min} \leq d_{\min} \leq 2p^s.$$

Proof: Let us first consider the case $p = 2$. We will show that the following set of *active* variable nodes in the Type II $\ell = 3$ LDPC constraint graph form a minimum-weight codeword: the root (variable) node, variable nodes $(x,0)$, $(0,0)$, $(1, \alpha^{p^s-2})$, (α, α^{p^s-3}), $(\alpha^2, \alpha^{p^s-4})$, $\ldots$, $(\alpha^i, \alpha^{p^s-2-i})$, $\ldots$, $(\alpha^{p^s-2}, 1)$ in layer L_2.

It is clear from this choice that there is exactly one active variable node from each class in layer L_2. Therefore, all the constraint nodes at layer L_1 are satisfied. The constraint nodes in the first class S'_0 of L_3 are $(0,0)'_c, (0,1)'_c, \ldots, (0, \alpha^{p^s-2})'_c$. The constraint node $(0,0)'_c$ is connected to $(x,0)$ and $(0,0)$, and the

constraint node $(0,\alpha^i)'_c$, for $i = 0, 1, 2, \ldots, p^s - 2$, is connected to variable nodes $(x, 0)$ and $(\alpha^i, \alpha^{p^s-2-i})$. Thus, all constraint nodes in S'_0 are satisfied. Let us consider the constraint nodes in class S'_{α^i}, for $i \in \{0, 1, 2, \ldots, p^s-2\}$. The variable node $(0, 0)$ connects to the constraint node $(\alpha^i, 0)'_c$ in S'_{α^i}. The variable node $(1, \alpha^{p^s-2})$ connects to the constraint node $(\alpha^i, \alpha^{p^s-2} + \alpha^i)'_c$ in S'_{α^i}, and in general, for $j = 0, 1, 2, \ldots, p^s - 2$, the variable node $(\alpha^j, \alpha^{p^s-2-j})$ connects to the constraint node $(\alpha^i, \alpha^{p^s-2-j} + \alpha^{i+j})'_c$ in S'_{α^i}. So enumerating all the constraint nodes in S'_{α^i}, with multiplicities, that are connected to an active variable node in L_2, we obtain

$(\alpha^i, 0)'_c$, $(\alpha^i, \alpha^{p^s-2} + \alpha^i)'_c$, $(\alpha^i, \alpha^{p^s-3} + \alpha^{i+1})'_c$, $\ldots$, $(\alpha^i, \alpha^{p^s-(p^s-i-1)} + \alpha^{p^s-3})'_c$, $(\alpha^i, \alpha^{p^s-(p^s-i)} + \alpha^{p^s-2})'_c$, $\ldots$.

Simplifying the exponents and rewriting this list, we see that, when i is odd, the constraint nodes are

$(\alpha^i, 0)'_c$, $(\alpha^i, \alpha^{p^s-2}+\alpha^i)'_c$, $(\alpha^i, \alpha^{p^s-3}+\alpha^{i+1})'_c$, $\ldots$, $(\alpha^i, \alpha^{i+1}+\alpha^{p^s-3})'_c$, $(\alpha^i, \alpha^i+\alpha^{p^s-2})'_c$, $(\alpha^i, \alpha^{i-1}+1)'_c$, $(\alpha^i, \alpha^{i-2}+\alpha)'_c$, $\ldots$, $(\alpha^i, \alpha^{(i-1)/2}+\alpha^{(i-1)/2})'_c$,$\ldots$, $(\alpha^i, \alpha+\alpha^{i-2})'_c$, $(\alpha^i, 1+\alpha^{i-1})'_c$.

(When i is even, the constraint nodes are $(\alpha^i, 0)'_c$, $(\alpha^i, \alpha^{p^s-2}+\alpha^i)'_c$, $(\alpha^i, \alpha^{p^s-3}+\alpha^{i+1})'_c$, $\ldots$, $(\alpha^i, \alpha^{(p^s-2+i)/2}+\alpha^{(p^s-2+i)/2})'_c$, $\ldots$, $(\alpha^i, \alpha^{i+1}+\alpha^{p^s-3})'_c$, $(\alpha^i, \alpha^i+\alpha^{p^s-2})'_c$, $(\alpha^i, \alpha^{i-1}+1)'_c$, $(\alpha^i, \alpha^{i-2}+\alpha)'_c$, $\ldots$, $(\alpha^i, \alpha^i+\alpha^{i-1})'_c$, $(\alpha^i, \alpha^{i-1}+\alpha^i)'_c$,$\ldots$, $(\alpha^i, 1+\alpha^{i-1})'_c$.) (Note that for $\beta \in GF(p^s)$, $\beta + \beta = 0$ when the characteristic of the field $GF(p^s)$ is two (i.e., $p = 2$).)

Observe that each of the constraint nodes in the above list appears exactly twice. Therefore, each constraint node in the list is connected to two active variable nodes in L_2, and hence, all the constraints in S'_{α^i} are satisfied. So we have that the set of $1+2^s+1$ active variable nodes forms a codeword. Furthermore, they must form a minimum-weight codeword since $d_{\min} \geq 2 + 2^s = T(2^s + 1, 6)$ by the tree bound of Theorem 1.2. This also proves that $d_{\min} = w_{\min} = T(d, 6)$ for $d = 2^s + 1$.

Now let us consider the case $p > 2$. The resulting codes are treated as p-ary codes. Consider the following set of active variable nodes: the root node, all but one of the nodes (x, y), for an appropriately chosen y, in class S_x, and the nodes (α^i, α^i), for $i = 0, 1, 2, \ldots, p^s - 2$. We have chosen $2p^s$ active variable nodes in all.

The variable nodes $(1, 1), (\alpha, \alpha), \ldots, (\alpha^{p^s-2}, \alpha^{p^s-2})$ are connected to constraint nodes

$$(\alpha^k, 0)'_c, (\alpha^k, 1 + \alpha^k)'_c, (\alpha^k, \alpha + \alpha^{k+1})'_c, \ldots, (\alpha^k, \alpha^j + \alpha^{k+j})'_c, \ldots, (\alpha^k, \alpha^i + \alpha^{k+p^s-2})'_c,$$

respectively, in class S'_{α^k} of constraint nodes in layer L_3. These nodes are either all distinct or all equal to $(\alpha^k, 0)'_c$ since $\alpha^r + \alpha^{k+r} = \alpha^t + \alpha^{k+t}$ if and only if either $r = t$ or $1+\alpha^k = 0$. Since $1+\alpha^k$ is zero for exactly one value of $k \in \{0, 1, 2, \ldots, p^s - 2\}$, we have that the variable nodes (α^i, α^i), for $i = 0, 1, 2, \ldots, p^s - 2$, are connected to distinct constraint nodes in all but one class $S_{\alpha^{k*}}$ and that, in $S_{\alpha^{k*}}$, they are all connected to the constraint node $(\alpha^{k*}, 0)'_c$. (Note that $k*$ satisfies $1 + \alpha^{k*} = 0$.) We let $y = \alpha^{k*}$. Therefore, the set of active variable nodes includes all nodes of the form (x, t), for $t = 0, 1, \alpha, \alpha^2, \ldots$, excluding node (x, y).

Since the code is p-ary, assign the following values to the chosen set of active variable nodes: assign the value 1 to the root variable node and to all the active variable nodes in class S_x, and assign the value $p-1$ to the active variable nodes (α^i, α^i), for $i = 0, 1, 2, \ldots, p^s - 2$. It is now easy to verify that all the constraints are satisfied.

Thus, $d_{\min} \leq 2p^s$. From Theorem 3.2 and Lemmas 2.4 and 2.6, we have $T(p^s+1, 6) \leq w_{\min} \leq d_{\min} \leq 2p^s$. ■

For degrees $d = p^s + 1$, $p > 2$, treating the Type II $\ell = 3$ LDPC constraint graphs as binary LDPC codes, yields $[n, 1, n]$ repetition codes, where $n = p^{2s} + p^s + 1$, $d_{\min} = n$, and dimension is 1. However, when the Type II $\ell = 3$ LDPC constraint graphs, for degrees $d = p^s + 1$, $p > 2$, are treated as p-ary LDPC codes, we believe that the distance $d_{\min} \geq p^s + 3$, and that this bound is in fact tight.

C. $\ell = 4$

For $d = p^s + 1$, p a prime and s a positive integer, a d regular tree T is enumerated from a root (variable) node for $\ell = 4$ layers L_0, L_1, L_2, L_3.

1) The nodes in L_0, L_1, and L_2 labeled as in the $\ell = 3$ case. The constraint nodes in L_3 are labeled as follows: The constraint nodes descending from variable node (x, j), for $j = 0, 1, , \alpha, \ldots, \alpha^{p^s-2}$, are labeled $(x, j, 0)_c, (x, j, 1)_c, \ldots, (x, j, \alpha^{p^s-2})_c$, the constraint nodes descending from variable node (i, j), for $i, j = 0, 1, , \alpha, \ldots, \alpha^{p^s-2}$, are labeled $(i, j, 0)_c, (i, j, 1)_c, \ldots, (i, j, \alpha^{p^s-2})_c$.
2) A final layer $L_\ell = L_4$ of $(d-1)^{\ell-1} = p^{3s}$ variable nodes is introduced. The p^{3s} variable nodes in L_4 are labeled as $(0, 0, 0)', (0, 0, 1)', \ldots, (0, 0, \alpha^{p^s-2})'$, $(0, 1, 0)', (0, 1, 1)', \ldots, (0, 1, \alpha^{p^s-2})'$, $\ldots$, $(\alpha^{p^s-2}, 0, 0)'$, $(\alpha^{p^s-2}, 0, 1)'$, $\ldots$, $(\alpha^{p^s-2}, 0, \alpha^{p^s-2})'$, $\ldots$, $(\alpha^{p^s-2}, \alpha^{p^s-2}, 0)'$, $(\alpha^{p^s-2}, \alpha^{p^s-2}, 1)'$, $\ldots$, $(\alpha^{p^s-2}, \alpha^{p^s-2}, \alpha^{p^s-2})'$. (Note that the '' in the labeling refers to nodes that are not in the tree T and the subscript 'c' refers to constraint nodes.)
3) For $0 \leq i \leq p^s - 1$, $0 \leq j \leq p^s - 1$, connect the constraint node $(x, i, j)_c$ to the variable nodes
$$(i, j, 0)', (i, j, 1)', (i, j, \alpha)', \ldots, (i, j, \alpha^{p^s-2})'.$$
4) To connect the remaining constraint nodes in L_3 to the variable nodes in L_4, we first define a function f. For $i, j, k, t = 0, 1, \alpha, \ldots, \alpha^{p^s-2}$ let
$$f : \mathbb{F} \times \mathbb{F} \times \mathbb{F} \times \mathbb{F} \to \mathbb{F}$$
$$(i, j, k, t) \mapsto y,$$
be an appropriately chosen function, that we will define later for some specific cases of the Type II $\ell = 4$ construction. Then, for $i, j, k = 0, 1, \alpha, \ldots, \alpha^{p^s-2}$, connect the constraint node $(i, j, k)_c$ in L_3 to the following variable nodes in L_4
$$(0, k+i\cdot 0, f(i, j, k, 0))',\ (1, k+i\cdot 1, f(i, j, k, 1))',\ (\alpha, k+i\cdot\alpha, f(i, j, k, \alpha))',\ ..,\ (\alpha^{p^s-2}, k+i\cdot\alpha^{p^s-2}, f(i, j, k, \alpha^{p^s-2}))'.$$
(Observe that the second index corresponds to the linear map defined by the array M_i defined in Section 2.B. Further, note that if $f(i, j, k, t) = j + i \cdot t$, then the resulting graphs obtained from the above set of connections have girth at least six. However, there are other functions $f(i, j, k, t)$ for which the resulting graphs have girth exactly eight, which is the best possible when $\ell = 4$ in this construction. At this point,

we do not have a closed form expression for the function f and we only provide details for specific cases below. (These cases were verified using the MAGMA software [10].)

The Type II, $\ell = 4$, LDPC codes have girth eight, minimum distance $d_{\min} \geq 2(p^s + 1)$, and blocklength $N = 1 + p^s + p^{2s} + p^{3s}$. (We believe that the tree bound on the minimum distance is met for most of the Type II, $\ell = 4$, codes, i.e. $d_{\min} = w_{\min} = 2(p^s + 1)$.) Figure 9 illustrates the general construction procedure. For $d = 3$, the Type II, $\ell = 4$, LDPC constraint graph as shown in Figure 10 corresponds to the $(2, 2)$-Finite-Generalized-Quadrangles-based LDPC (FGQ LDPC) code of [18]; the function f used in constructing this example is defined by $f(i, j, k, t) = j + (i + 1) \cdot t$, i.e., the map defined by the array M_{i+1}. The orthogonal arrays used for constructing this code are

$$M_0 = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \; M_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

We now state some results concerning the choice of the function f.

1) The Type II $\ell = 4$ construction results in incidence graphs of finite generalized quadrangles for appropriately chosen functions f. These graphs have girth 8 and diameter 4.
2) For some specific cases, examples of the function $f(i, j, k, t)$ that resulted in a girth 8 graph is given in Table II. (Note that for the second entry in the table, the function $g : GF(4) \to GF(4)$ is defined by the following maps: $0 \mapsto 1$, $1 \mapsto \alpha$, $\alpha \mapsto \alpha^2$, and $\alpha^2 \mapsto 0$.) We have not been able to find a general relation or a closed form expression for f yet.

p	s	elements of $GF(p^s)$	degree $d = p^s + 1$	$f(i, j, k, t)$
2	1	$\{0, 1\}$	3	$j + (i + 1) \cdot t$
2	2	$\{0, 1, \alpha, \alpha^2\}$	5	$j + g(i) \cdot t$
3	1	$\{0, 1, 2\}$	4	$i \cdot (k + 2 \cdot i \cdot t) + j$
3	2	$\{0, 1, \alpha, .., \alpha^7\}$	10	$i \cdot (k + \alpha \cdot i \cdot t) + j$
5	1	$\{0, 1, 2, 3, 4\}$	6	$i \cdot (k + 3 \cdot i \cdot t) + j$
7	1	$\{0, 1, 2, .., 6\}$	8	$i \cdot (k + 4 \cdot i \cdot t) + j$

TABLE II

THE FUNCTION f FOR THE TYPE II $\ell = 4$ CONSTRUCTION.

3) For the above set of functions, the resulting Type II $\ell = 4$ LDPC constraint graphs have minimum distance meeting the tree bound, when $p = 2$, i.e., $d_{\min} = w_{\min} = 2(2^s + 1)$. We conjecture that, in general, for degrees $d = 2^s + 1$, the Type II $\ell = 4$ girth eight LDPC constraint graphs have $d_{\min} = w_{\min} = T(2^s + 1, 8) = 2(2^s + 1)$.
4) For degrees $d = p^s + 1$, $p > 2$, we expect the corresponding p-ary LDPC codes from this construction to have minimum distances $d_{\min}$ either equal or very close to the tree bound. Hence, we also expect the corresponding minimum pseudocodeword weight $w_{\min}$ to be close to $d_{\min}$.

The above results were verified using MAGMA and computer simulations.

D. Remarks

It is well known in the literature that finite generalized polygons (or, N-gons) of order p^s exist [11]. A finite generalized N-gon is a non-empty point-line geometry, and consists of a set $\mathcal{P}$ of points and a set $\mathcal{L}$ of lines such that the incidence graph of this geometry is a bipartite graph of diameter N and girth $2N$. Moreover, when each point is incident on $t+1$ lines and each line contains $t+1$ points, the order of the N-gon is said be to t. The Type II $\ell = 3$ and $\ell = 4$ constructions yield finite generalized 3-gons and 4-gons, respectively, of order p^s. These are essentially finite projective planes and finite generalized quadrangles. The Type II construction can be similarly extended to larger ℓ. We believe that finding the right connections for connecting the nodes between the last layer in T and the final layer will yield incidence graphs of these other finite generalized polygons. For instance, for $\ell = 6$ and $\ell = 8$, the construction can yield finite generalized hexagons and finite generalized octagons, respectively. We conjecture that the incidence graphs of generalized N-gons yield LDPC codes with minimum pseudocodeword weight $w_{\min}$ very close to the corresponding minimum distance $d_{\min}$ and particularly, for generalized N-gons of order 2^s, the LDPC codes have $d_{\min} = w_{\min} = T(2^s + 1, 2N)$.

IV. Simulation Results

A. Performance with min-sum iterative decoding

Figures 12, 13, 14, 15 show the bit-error-rate performance of Type I-A, Type I-B, Type II $\ell = 3$ (girth six), and Type II $\ell = 4$ (girth eight) LDPC codes, respectively, over the binary input additive white Gaussian noise channel (BIAWGNC) with min-sum (not, sum-product!) iterative decoding. The performance of regular or semi-regular randomly constructed LDPC codes of comparable rates and blocklengths are also shown. (All of the random LDPC codes compared in this paper have a variable node degree of three and are constructed from the online LDPC software available at `http://www.cs.toronto.edu/~radford/ldpc.software.html`.)

Figure 12 shows that the Type I-A LDPC codes perform substantially better than their random counterparts. Figure 13 reveals that the Type I-B LDPC codes perform better than comparable random LDPC codes at short blocklengths; but as the blocklengths increase, the random LDPC codes tend to perform better in the waterfall region. Eventually however, as the SNR increases, the Type I-B LDPC codes outperform the random ones and, unlike the random codes, they do not have a prominent error floor. Figure 14 reveals that the performance of Type II $\ell = 3$ (girth-six) LDPC codes is also significantly better than comparable random codes; these codes correspond to the two dimensional PG-LDPC codes of [15]. Figure 15 indicates the performance of Type II $\ell = 4$ (girth-eight) LDPC codes; these codes perform comparably to random codes at short blocklengths, but at large blocklengths, the random codes perform better as they have larger relative minimum distances compared to the Type II $\ell = 4$ (girth-eight) LDPC codes.

As a general observation, min-sum iterative decoding of most of the tree-based LDPC codes (particularly, Type I-A, Type II, and some Type I-B) presented here did not typically reveal detected errors, i.e., errors caused due to the decoder failing to converge to any valid codeword within the maximum specified number of iterations, which was set to 200 in these simulations. Detected errors are caused primarily due to the presence of

pseudocodewords, especially those of minimum weight. We think that the relatively low occurrences of detected errors with iterative decoding of these LDPC codes is primarily due to their good[3] minimum pseudocodeword weight $w_{\min}$.

B. Performance of Type I-B and Type II LDPC codes with sum-product iterative decoding

Figures 16, 17, and 18 show the performance of the Type I-B, Type II $\ell = 3$ and Type II $\ell = 4$, respectively, LDPC codes with sum-product iterative decoding for the BIAWGNC. The performance is shown only for a few codes from each construction. The main observation from these performance curves is that the the tree-based LDPC codes perform relatively much better than random LDPC codes of comparable parameters when the decoding is sum-product instead of the min-sum algorithm. Although the Type I-B LDPC codes perform a little inferior to their random counterparts in the waterfall region, the gap between the performances of the random and the Type I-B LDPC codes is much smaller with sum-product decoding than with min-sum decoding. (Compare Figures 13 and 16.) Similarly, comparing Figures 17 and 14, we see that the Type II $\ell = 3$ LDPC codes perform relatively much better than their random counterparts with sum-product decoding than with min-sum decoding. Figure 18, in comparison with Figure 15, shows a similar trend in performance of Type II $\ell = 4$ (girth 8) LDPC codes with sum-product iterative decoding.

Note that the simulation results for the min-sum and sum-product decoding correspond to the case when the LDPC codes resulting from constructions Type I and Type II were treated as binary LDPC codes for all choices of degree $d = p^s$ or $d = p^s + 1$. We will now examine the performance when the codes are treated as p-ary codes if the corresponding degree in the LDPC constraint graph is $d = p^s$ (for Type I-B) or $d = p^s + 1$ (for Type II). (Note that this will affect only the performances of those codes for which p is not equal to two.)

C. Performance of p-ary Type I-B and Type II LDPC codes over the p-ary symmetric channel

We examine the performance of the p-ary LDPC codes obtained from the Type I-B and Type II constructions on the p-ary symmetric channel instead of the AWGN channel. The p-ary symmetric channel is shown in Figure 11. An error occurs with probability ϵ, the channel transition probability. Figures 19, 20, and 21 show the performance of Type I-B, Type II $\ell = 3$ and Type II $\ell = 4$, 3-ary LDPC codes, respectively, on the 3-ary symmetric channel with sum-product iterative decoding. The parity check matrices resulting the the Type I-B and Type II constructions are considered to be matrices over the field $GF(3)$ and sum-product iterative decoding is implemented as outlined in [3]. The corresponding plots show the information symbol error rate as a function of the channel transition probability ϵ. In Figure 19, the performance of 3-ary Type I-B LDPC codes obtained for degrees $d = 3$, $d = 3^2$, and $d = 3^3$, is shown and compared with the performance of random 3-ary LDPC codes of comparable rates and block lengths. (To make a fair comparison, the random

[3]i.e., relative to the minimum distance $d_{\min}$.

LDPC codes also have only zeros and ones as entries in their parity check matrices. It has been observed in [3] that choosing the non-zero entries in the parity check matrices of non-binary codes cleverly can yield some performance gain, but this avenue was not explored in these simulations.) In Figure 20, the performance of 3-ary Type II $\ell = 3$ (girth six) LDPC codes obtained for degrees $d = 3+1$, $d = 3^2+1$, and $d = 3^3+1$, is shown and compared with random 3-ary LDPC codes. Figure 21 shows the analogous performance of 3-ary Type II $\ell = 4$ (girth eight) LDPC codes obtained for degrees $d = 3+1$ and $d = 3^2+1$. In all these plots, it is seen that the tree-based constructions perform comparably or better than random LDPC codes of similar rates and block lengths. (In some cases, the performance of the tree-based constructions is significantly better than that of random LDPC codes (example, Figure 21).)

The simulation results show that the tree-based constructions yield a wide range of LDPC codes that perform very well with iterative decoding.

V. Conclusions

The Type I construction yields a family of LDPC codes that, to the best of our knowledge, do not correspond to any of the LDPC codes obtained from finite geometries or other geometrical objects. It would be interesting to extend the Type II construction to more layers as described at the end of Section 4. The two tree-based constructions presented in this paper yield a wide range of codes that perform well when decoded iteratively, largely due to the maximized minimum pseudocodeword weight. However, the overall minimum distance of the code is relatively small. Constructing codes with larger minimum distance, while still maintaining $d_{\min} = w_{\min}$, remains a challenging problem.

Appendix

A. Pseudocodeword weight for p-ary LDPC codes on the p-ary symmetric channel

Suppose the all-zero codeword is sent across a p-ary symmetric channel and the vector $\mathbf{r} = (r_0, r_1, \ldots, r_{n-1})$ is received. Then errors occur in positions where $r_i \neq 0$. Let $S = \{i|\ r_i \neq 0\}$ and let $S^c = \{i|\ r_i = 0\}$. The distance between $\mathbf{r}$ and a pseudocodeword F is defined as

$$d(\mathbf{r}, F) = \sum_{i=0}^{n-1} \sum_{k=0}^{p-1} \chi(r_i \neq k) f_{i,k}, \tag{2}$$

where $\chi(P)$ is an indicator function that is equal to 1 if the proposition P is true and is equal to 0 otherwise.

The distance between $\mathbf{r}$ and the all-zero codeword $\mathbf{0}$ is

$$d(\mathbf{r}, \mathbf{0}) = \sum_{i=0}^{n-1} \chi(r_i \neq 0)$$

which is the Hamming weight of $\mathbf{r}$ and can be obtained from equation (2).

The iterative decoder chooses in favor of F instead of the all-zero codeword $\mathbf{0}$ when $d(\mathbf{r}, F) \leq d(\mathbf{r}, \mathbf{0})$. That is, if

$$\sum_{i \in S^c} (1 - f_{i,0}) + \sum_{i \in S} (1 - f_{i,r_i}) \leq \sum_{i \in S} 1$$

The condition for choosing F over the all-zero codeword reduces to

$$\Big\{ \sum_{i \in S^c} (1 - f_{i,0}) \leq \sum_{i \in S} f_{i,r_i} \Big\}$$

Hence, we define the weight of a pseudocodeword F in the following manner.

Let e be a number such that the sum of the e largest components in the matrix F', say, $f_{i_1,j_1}, f_{i_2,j_2}, \ldots, f_{i_e,j_e}$, exceeds $\sum_{i \neq i_1,i_2,\ldots,i_e} (1 - f_{i,0})$. Then the weight of F on the p-ary symmetric channel is defined as

$$w_{PSC}(F) = \begin{cases} 2e, & \text{if } f_{i_1,j_1} + \cdots + f_{i_e,j_e} = \sum_{i \neq i_1,i_2,\ldots,i_e} (1 - f_{i,0}) \\ 2e - 1, & \text{if } f_{i_1,j_1} + \cdots + f_{i_e,j_e} > \sum_{i \neq i_1,i_2,\ldots,i_e} (1 - f_{i,0}) \end{cases}$$

Note that in the above definition, none of the j_k's, for $k = 1, 2, \ldots, e$, are equal to zero, and all the i_k's, for $k = 1, 2, \ldots, e$, are distinct. That is, we choose at most one component in every column of F' when picking the e largest components. The received vector $\mathbf{r} = (r_0, r_1, \ldots, r_{n-1})$ that has the following components: $r_{i_1} = j_1, r_{i_2} = j_2, \ldots, r_{i_e} = j_e$, $r_i = 0$, for $i \notin \{i_1, i_2, \ldots, i_e\}$, will cause the decoder to make an error and choose F over the all-zero codeword.

B. Proof of Lemma 2.6

Proof:

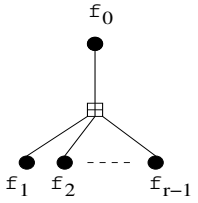


Fig. 1. Single constraint code.

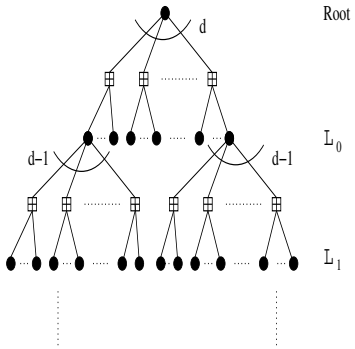


Fig. 2. Local tree structure for a d-left regular graph.

$$(1 - f_{i,0}) \leq \textstyle\sum_{j \neq i} (1 - f_{j,0})$$

$$\begin{aligned} d(1 - f_{0,0}) &\leq \textstyle\sum_{j \in L_0} (1 - f_{j,0}), \\ d(d-1)(1 - f_{0,0}) &\leq \textstyle\sum_{j \in L_1} (1 - f_{j,0}) \\ &\vdots \end{aligned}$$

Case: $\frac{g}{2}$ odd. Consider a single constraint node with r variable node neighbors as shown in Figure 1. Then, for $i = 0, 1, \ldots, r-1$ and $k = 0, 1, \ldots, p-1$, the following inequality holds:

$$f_{i,k} \leq \sum_{j \neq i} \sum_{\sigma_j : \sum \sigma_j + k = 0 \bmod p} \frac{f_{j,\sigma_j} \sigma_j}{\sum \sigma_j}$$

where the middle summation is over all possible assignments $\sigma_j \in \{0, 1, \ldots, p-1\}$ to the variable nodes $j \neq i$ such that $k + \sum_{j \neq i} \sigma_j = 0 \bmod p$, i.e., this is a valid assignment for the constraint node. The innermost summation in the denominator is over all $j \neq i$.

However, for $i = 0, 1, \ldots, r-1$, the following (weaker) inequality also holds:

$$(1 - f_{i,0}) \leq \sum_{j \neq i} (1 - f_{j,0}) \tag{3}$$

Now let us consider a d-left regular LDPC constraint graph representing a p-ary LDPC code. We will enumerate the LDPC constraint graph as a tree from an arbitrary root variable node, as shown in Figure 2. Let F be a pseudocodeword matrix for this graph. Without loss of generality, let us assume that the component $(1 - f_{0,0})$ corresponding to the root node is the maximum among all $(1 - f_{i,0})$ over all i.

Applying the inequality in (3) at every constraint node in first constraint node layer of the tree, we obtain

$$d(1 - f_{0,0}) \leq \sum_{j \in L_0} (1 - f_{j,0}),$$

where L_0 corresponds to variable nodes in first level of the tree. Subsequent application of the inequality in (3) to the second layer of constraint nodes in the tree yields

$$d(d-1)(1 - f_{0,0}) \leq \sum_{j \in L_1} (1 - f_{j,0}),$$

Continuing this process until layer $L_{\frac{g-6}{4}}$, we obtain

$$d(d-1)^{\frac{g-6}{4}}(1 - f_{0,0}) \leq \sum_{j \in L_{\frac{g-6}{4}}} (1 - f_{j,0})$$

Since the LDPC graph has girth g, the variable nodes up to level $L_{\frac{g-6}{4}}$ are all distinct. The above inequalities yield:

$$[1 + d + d(d-1) + \cdots + d(d-1)^{\frac{g-6}{4}}](1 - f_{0,0}) \leq \sum_{i \in \{0\} \cup L_0 \cup \ldots L_{\frac{g-6}{4}}} (1 - f_{i,0}) \leq \sum_{\text{all } i} (1 - f_{i,0}) \tag{4}$$

Let e the smallest number such that there are e maximal components f_{i_1,j_1}, $f_{i_2,j_2}, f_{i_3,j_3}, \ldots, f_{i_e,j_e}$, for $i_1, i_2, \ldots, i_e$ all distinct and $j_1, j_2, \ldots, j_e \in \{1, 2, \ldots, p-1\}$, in F' (the sub-matrix of F excluding the first row in F) such that

$$f_{i_1,j_1} + f_{i_2,j_2} + \cdots + f_{i_e,j_e} \geq \sum_{i \notin \{i_1, i_2, i_3, \ldots, i_e\}} (1 - f_{i,0})$$

Then, since none of the j_k's, $k = 1, 2, \ldots, e$, are zero, we clearly have

$$(1 - f_{i_1,0}) + (1 - f_{i_2,0}) + \cdots + (1 - f_{i_e,0}) \geq f_{i_1,j_1} + f_{i_2,j_2} + \cdots + f_{i_e,j_e} \geq \sum_{i \notin \{i_1, i_2, i_3, \ldots, i_e\}} (1 - f_{i,0})$$

Hence we have that

$$2((1 - f_{i_1,0}) + (1 - f_{i_2,0}) + \cdots + (1 - f_{i_e,0})) \geq \sum_{\text{all } i} (1 - f_{i,0})$$

We can then lower bound this further using the inequality in (4) as

$$2((1-f_{i_1,0})+(1-f_{i_2,0})+\cdots+(1-f_{i_e,0})) \geq [1+d+d(d-1)+\cdots+d(d-1)^{\frac{g-6}{4}}](1-f_{0,0})$$

Since we assumed that $(1-f_{0,0})$ is the maximum among $(1-f_{i,0})$ over all i, we have

$$2e(1-f_{0,0}) \geq 2((1-f_{i_1,0})+(1-f_{i_2,0})+\cdots+(1-f_{i_e,0})) \geq [1+d+d(d-1)+\cdots+d(d-1)^{\frac{g-6}{4}}](1-f_{0,0})$$

This yields the desired bound

$$w_{PSC}(F) = 2e \geq 1+d+d(d-1)+\cdots+d(d-1)^{\frac{g-6}{4}}$$

Since the pseudocodeword F was arbitrary, we also have $w_{min} \geq 1+d+d(d-1)+\cdots+d(d-1)^{\frac{g-6}{4}}$. The case $\frac{g}{2}$ even is treated similarly. ■

C. TABLE OF CODE PARAMETERS

The code parameters resulting from the tree-based constructions are summarized in Tables III, IV, V, and VI. Note that * indicates an upper bound instead of the exact minimum distance (or minimum pseudocodeword weight) since it was computationally hard to find the distance (or pseudoweight) for those cases. Similarly, for cases where it was computationally hard to get any reasonable bound the minimum pseudocodeword weight, the corresponding entry in the table is left empty. The lower bound on $w_{\min}$ seen in the tables corresponds to the tree bound (Theorem 1.2). It is observed that when the codes resulting from the construction are treated as p-ary codes rather than binary codes when the corresponding degree in the LDPC graph is $d = p^s$ (for Type I-B) or $d = p^s + 1$ (for Type II), the resulting rates obtained are much superior; we also believe that the minimum pseudocodeword weights (on the p-ary symmetric channel) are much closer to the minimum distances for these p-ary LDPC codes.

REFERENCES

[1] N. L. Biggs. Construction for cubic graphs of large girths. *Electronic Journal of Combinatorics*, 5, 1998.

[2] R. C. Bose. On the application of the properties of Galois fields to the problem of construction of Hyper-Graceo-Latin squares. *Sankhyā*, pages 323–338, 1938.

[3] M. C. Davey. *Error-correction using Low-Density Parity-Check Codes*. PhD thesis, University of Cambridge, 1999.

[4] M. C. Davey and D. J. C. MacKay. Low density parity check codes over GF(q). *IEEE Communications Letters*, 2(6):159–166, June 1998.

[5] G. D. Forney, Jr., R. Koetter, F. Kschishang, and A. Reznik. On the effective weights of pseudocodewords for codes defined on graphs with cycles. In B. Marcus and J. Rosenthal, editors, *Codes, Systems and Graphical Models*, IMA Vol. 123, pages 101–112. Springer-Verlag, 2001.

[6] C. Kelley and D. Sridhara. Pseudocodewords of Tanner graphs. Submitted to IEEE Trans. on Information Theory, June 2005.

[7] C. Kelley, D. Sridhara, J. Xu, and J. Rosenthal. Pseudocodeword Weights and Stopping Sets. In *Proc. of the IEEE International Symposium on Information Theory, (Chicago, USA)*, page 150, 2004.

[8] R. Koetter and P. Vontobel. Graph-covers and iterative decoding of finite length codes. In *Proceedings of the IEEE International Symposium on Turbo Codes and Applications*, Brest, France, September 2003.

[9] Y. Kou, S. Lin, and M. P. C. Fossorier. Low-density parity-check codes based on finite geometries: a rediscovery and new results. *IEEE Trans. Inform. Theory*, 47(7):2711–2736, 2001.

[10] MAGMA Software. Documentation available at http://magma.maths.usyd.edu.

[11] H. van Maldeghem. *Generalized Polygons*. Birkhäuser Verlag, Basel, Boston, Berlin, 1998.

[12] Fred S. Roberts. *Applied Combinatorics*. Prentice Hall Inc., Englewood Cliffs, NJ, 1984.

[13] L. Rudolph. A class of majority logic decodable codes. *IEEE Trans. Inform. Theory*, IT-13(2):305–307, April 1967.

[14] D. Sridhara and T. E. Fuja. LDPC codes over rings for PSK-modulation. *IEEE Trans. Inform. Theory*, IT-51(9):3209–3220, 2005.

[15] H. Tang, J. Xu, S. Lin, and K. A. S. Abdel-Ghaffar. Codes on finite geometries. *IEEE Trans. Inform. Theory*, IT-51(2):572–596, 2005.

[16] R. M. Tanner. A recursive approach to low complexity codes. *IEEE Trans. Inform. Theory*, 27(5):533–547, 1981.

[17] J. H. van Lint and R. M. Wilson. *A Course in Combinatorics*. Cambridge University Press, second edition, 2001.

[18] P. O. Vontobel and R. M. Tanner. Construction of codes based on finite generalized quadrangles for iterative decoding. In *Proceedings of the 2001 IEEE International Symposium on Information Theory*, page 223, Washington, D.C., 2001.

[19] N. Wiberg. *Codes and Decoding on General Graphs*. PhD thesis, Linköping University, Sweden, 1996.

No. of layers in T ℓ	block length n	degree $d=3$	dimension	rate	$d_{\min}$	$w_{\min}$	tree lower-bound	girth g	diameter δ
3	10	3	4	0.4000	4	4	4	6	5
4	22	3	4	0.1818	8	8^*	6	8	7
5	46	3	10	0.2173	10	10	10	10	9
6	94	3	14	0.1489	20	20^*	18	12	11
7	**190**	3	25	0.1315	24	≥ 18	18	**12**	13

TABLE III

SUMMARY OF TYPE I-A CODE PARAMETERS.

p	s	block length $n=p^{2s}+1$	degree $d=p^s$	dimension	rate	$d_{\min}$	$w_{\min}$	tree lower-bound	girth g	diameter δ	code alphabet
2	1	5	2	1	0.2000	5	5	4	8	4	binary
3	1	10	3	3	.3000	4	4	4	6	5	binary
				(2)	(0.2000)	(6)	( )				3-ary
2	2	17	4	5	0.2941	6	6^*	5	6	5	binary
5	1	26	5	7	0.2692	8	8^*	6	6	5	binary
				(7)	(0.2692)	(10)	( )				5-ary
7	1	50	7	11	0.2200	12	12^*	8	6	5	binary
				(16)	(0.3200)	(15)	( )				7-ary
2	3	65	8	31	0.4769	10	10^*	9	6	5	binary
3	2	82	9	15	0.1829	16	≥ 10	10	6	5	binary
				(38)	(0.4634)	(15)	( )				3-ary
11	1	122	11	19	0.1557	20^*	≥ 12	12	6	5	binary
				(46)	(0.3770)	(30^*)	( )				11-ary
2	4	257	16	161	0.6264	20^*	≥ 17	17	6	5	binary
5	2	626	25	47	0.075	48^*	≥ 26	26	6	5	binary
				(377)	(0.6022)	(90^*)	( )				5-ary
3	3	730	27	51	0.0698	52^*	≥ 28	28	6	5	binary
				(488)	(0.6684)	(60^*)	( )				3-ary
2	5	1025	32	751	0.7326	40^*	≥ 33	33	6	5	binary
7	2	2404	49	95	0.0395	96^*	≥ 50	50	6	5	binary
				(1572)	(0.6536)	(216^*)	( )				7-ary

TABLE IV

SUMMARY OF TYPE I-B CODE PARAMETERS.

p	s	block length $n = 1 + p^s + p^{2s}$	degree $d = p^s + 1$	dimension	rate	$d_{\min}$	$w_{\min}$	tree lower-bound	girth g	diameter δ	code alphabet
2	1	7	3	3	0.4285	4	4	4	6	3	binary
3	1	13	4	1	0.0769	13	6^*	5	6	3	binary
				(6)	(0.4615)	(6)	( )				3-ary
2	2	21	5	11	0.5238	6	6	6	6	3	binary
5	1	31	6	1	0.0322	31	10^*	7	6	3	binary
				(15)	(0.4838)	(10^*)	( )				5-ary
7	1	57	8	1	0.017	57	16^*	9	6	3	binary
				(28)	(0.4912)	(14^*)	( )				7-ary
2	3	73	9	45	0.6164	10	10	10	6	3	binary
3	2	91	10	1	0.0109	91	≥ 11	11	6	3	binary
				(54)	(0.5934)	(15^*)	( )				3-ary
2	4	273	17	191	0.6996	18	18	18	6	3	binary
5	2	651	26	1	0.0015	651	≥ 27	27	6	3	binary
				(425)	(0.6528)	(56^*)	( )				5-ary

TABLE V

SUMMARY OF TYPE II, $\ell = 3$ CODE PARAMETERS.

p	s	block length n=1+p^s+p^{2s}+p^{3s}	degree d=p^s+1	dimension	rate	$d_{\min}$	$w_{\min}$	tree lower-bound	girth g	diameter δ	code alphabet
2	1	15	3	5	0.3333	6	6	6	8	4	binary
3	1	40	4	15	0.3750	10	≥ 8	8	8	4	binary
				(15)	(0.3750)	(10^*)	( )				3-ary
2	2	85	5	35	0.4117	10	10	10	8	4	binary
5	1	156	6	65	0.4167	20^*	≥ 12	12	8	4	binary
				(65)	(0.4167)	(26^*)	( )				5-ary
7	1	400	8	175	0.4375	66^*	≥ 16	16	8	4	binary
				(175)	(0.4375)	(119^*)	( )				7-ary
2	3	585	9	287	0.4905	18	18	18	8	4	binary
3	2	820	10	369	0.4500	112^*	≥ 20	20	8	4	binary
				(395)	(0.4817)	(96^*)	( )				3-ary

TABLE VI

SUMMARY OF TYPE II, $\ell = 4$ CODE PARAMETERS.

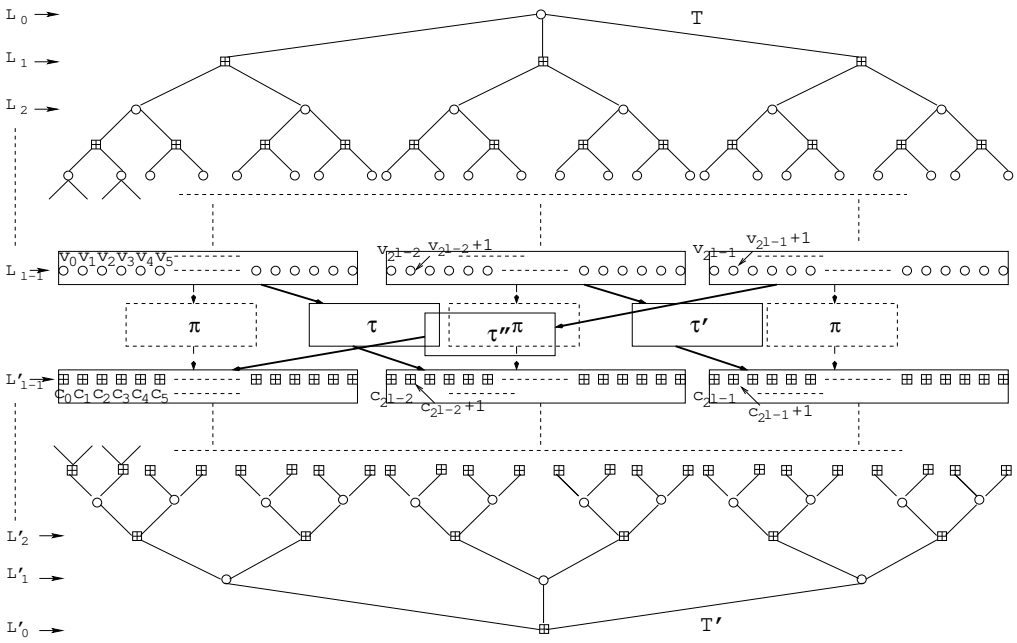


Fig. 3. Tree construction of Type I-A LDPC code.

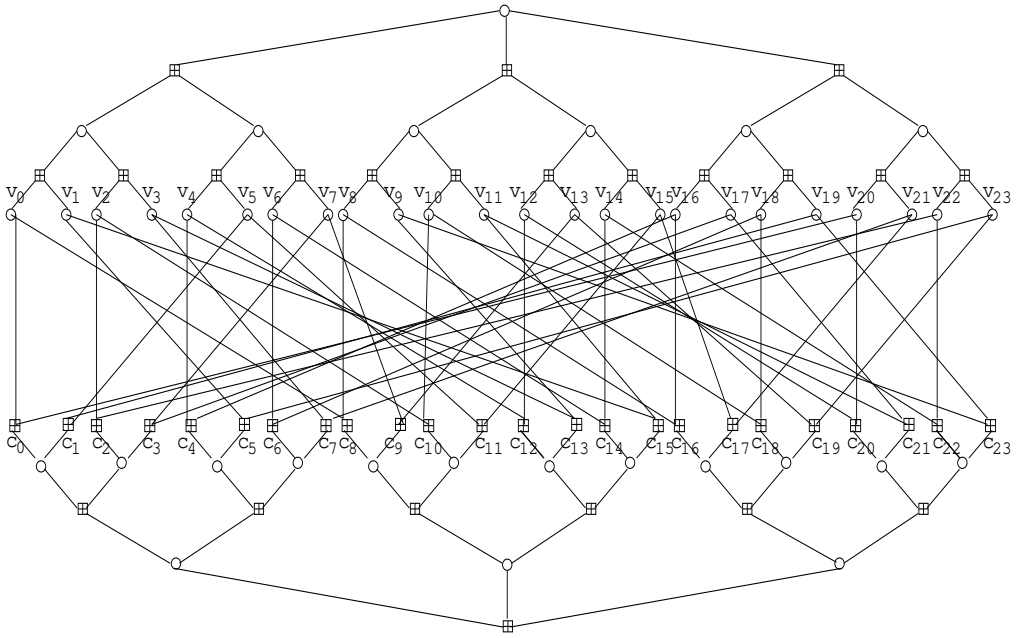


Fig. 4. Type I-A LDPC constraint graph having degree $d = 3$ and girth $g = 10$.

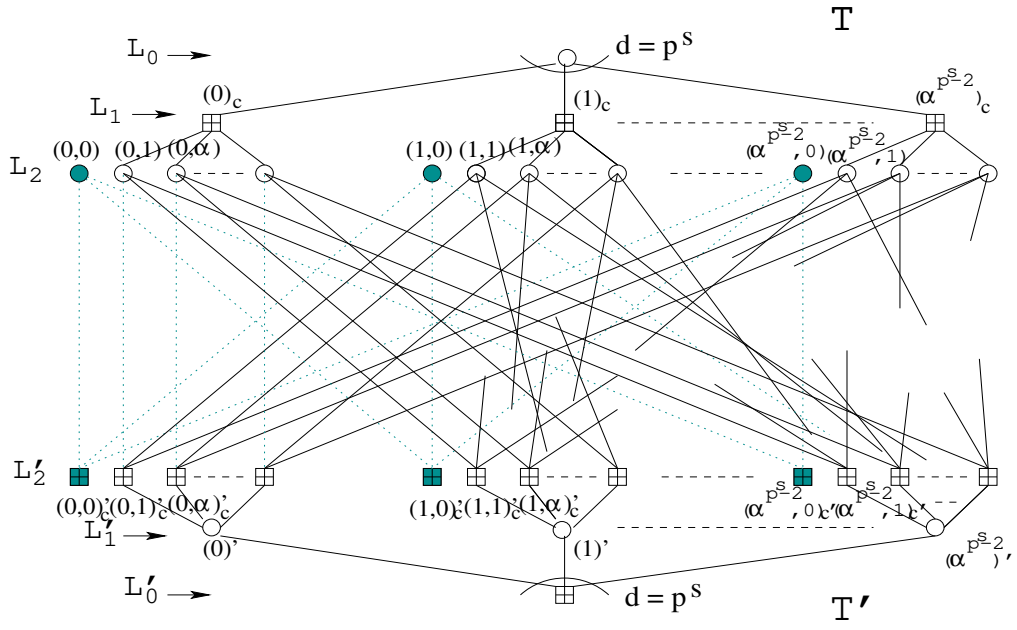


Fig. 5. Tree construction of Type I-B LDPC code. (Shaded nodes are imaginary nodes and dotted lines are imaginary lines.)

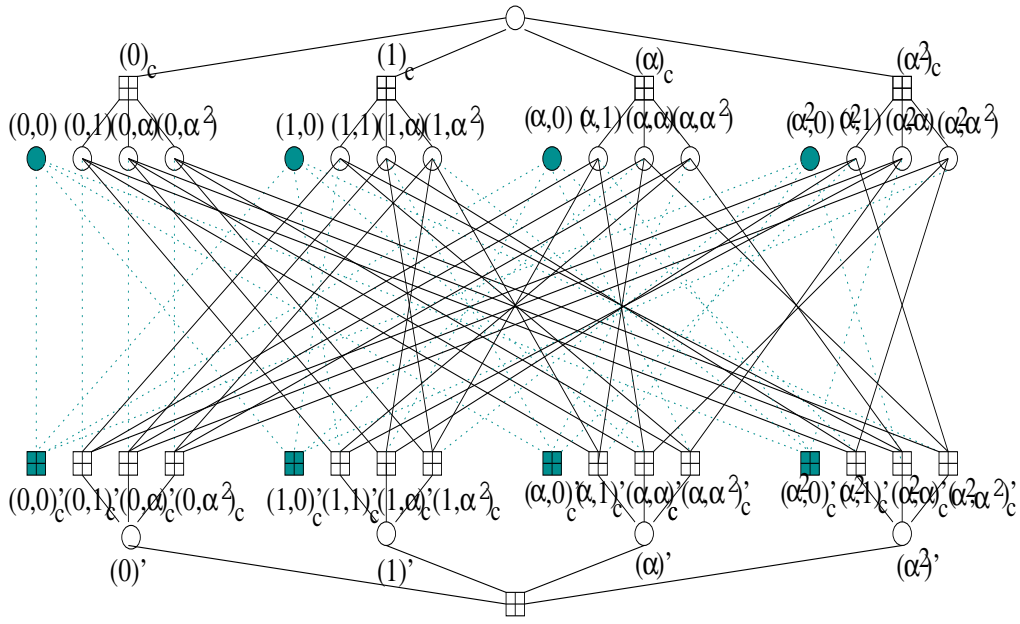


Fig. 6. Type I-B LDPC constraint graph having degree $d = 4$ and girth $g = 6$.

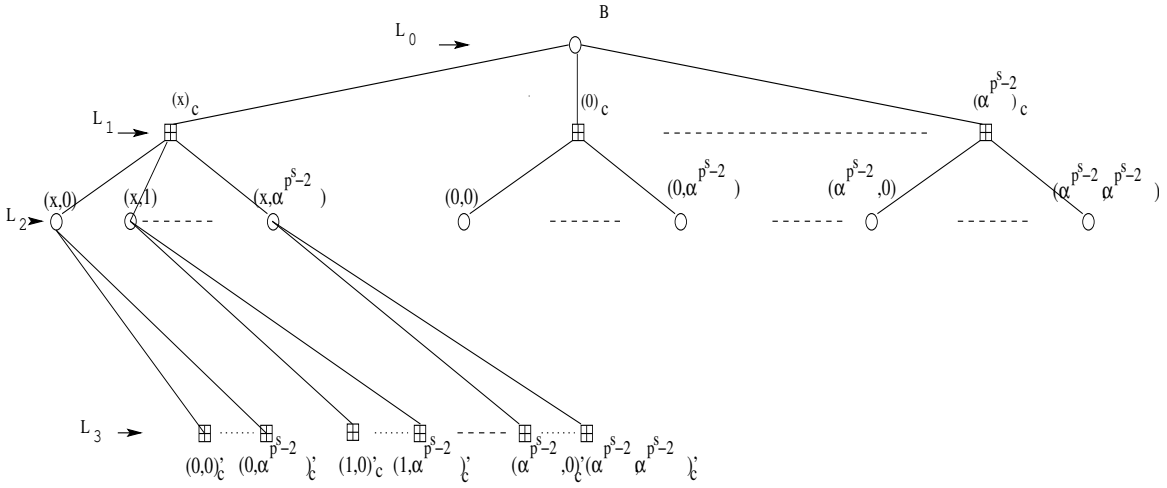


Fig. 7. Tree construction of girth 6 Type II ($\ell = 3$) LDPC code.

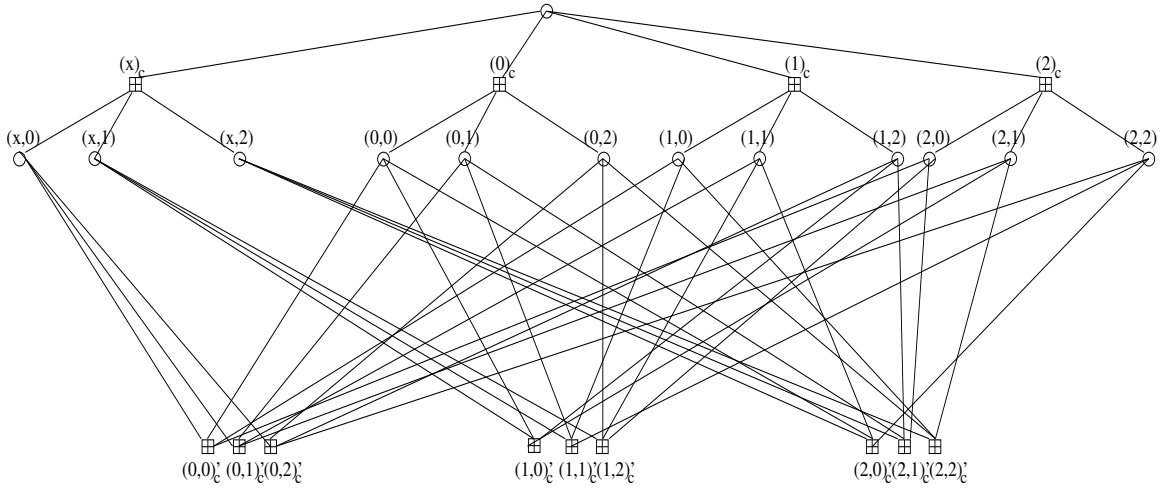


Fig. 8. Type II LDPC constraint graph having degree $d = 4$ and girth $g = 6$.

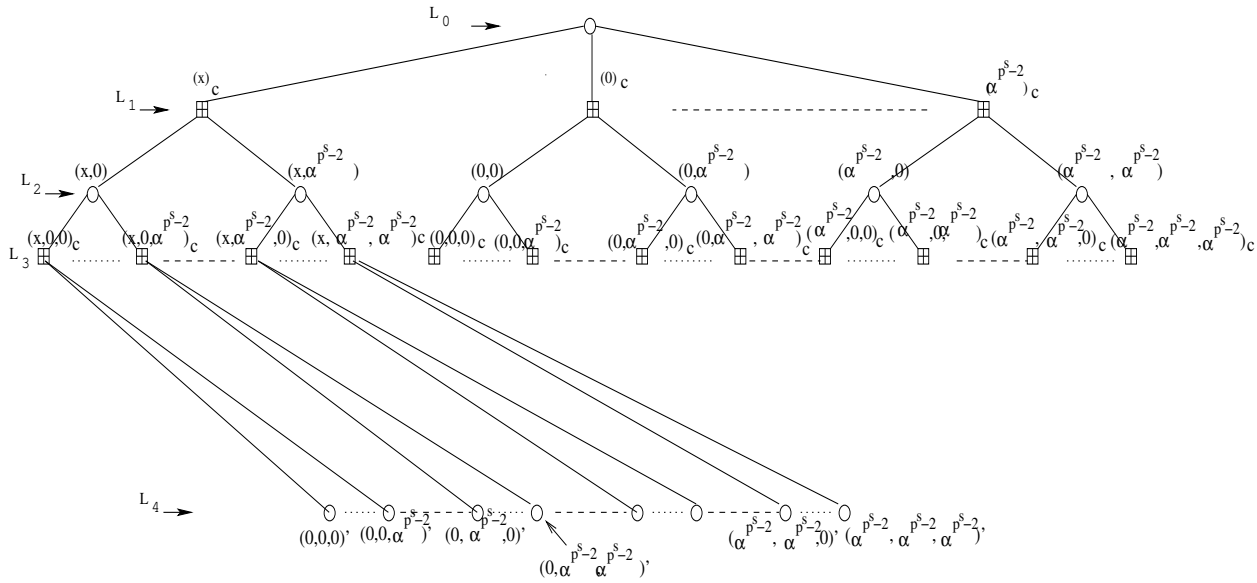


Fig. 9. Tree construction of girth 8 Type II ($\ell = 4$) LDPC code.

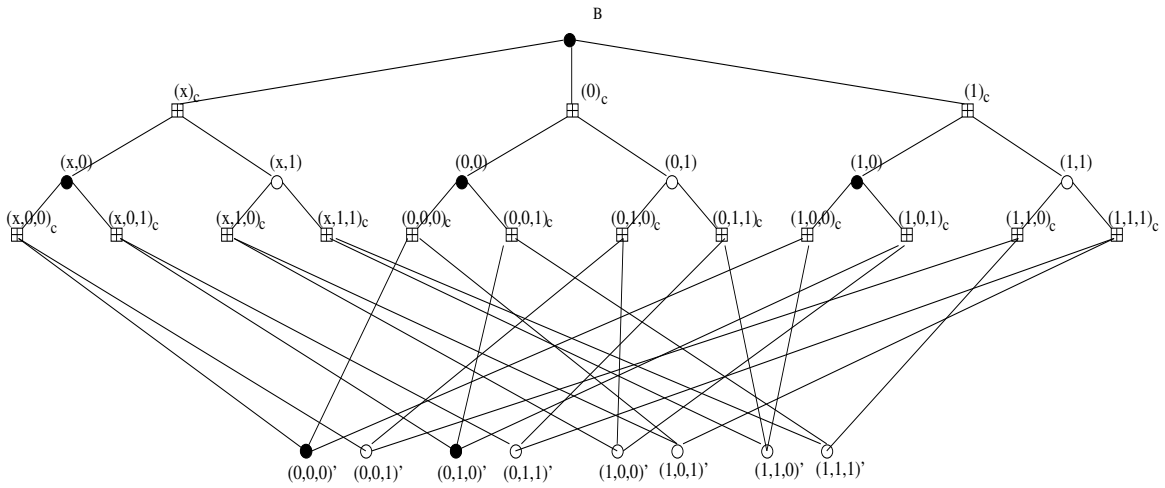


Fig. 10. Type II LDPC constraint graph having degree $d = 3$ and girth $g = 8$. (Shaded nodes highlight a minimum weight codeword.)

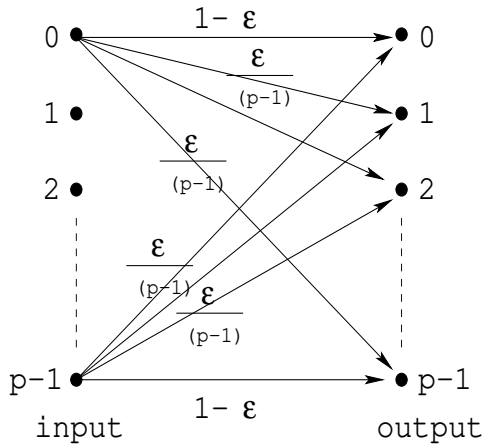


Fig. 11. A p-ary symmetric channel.

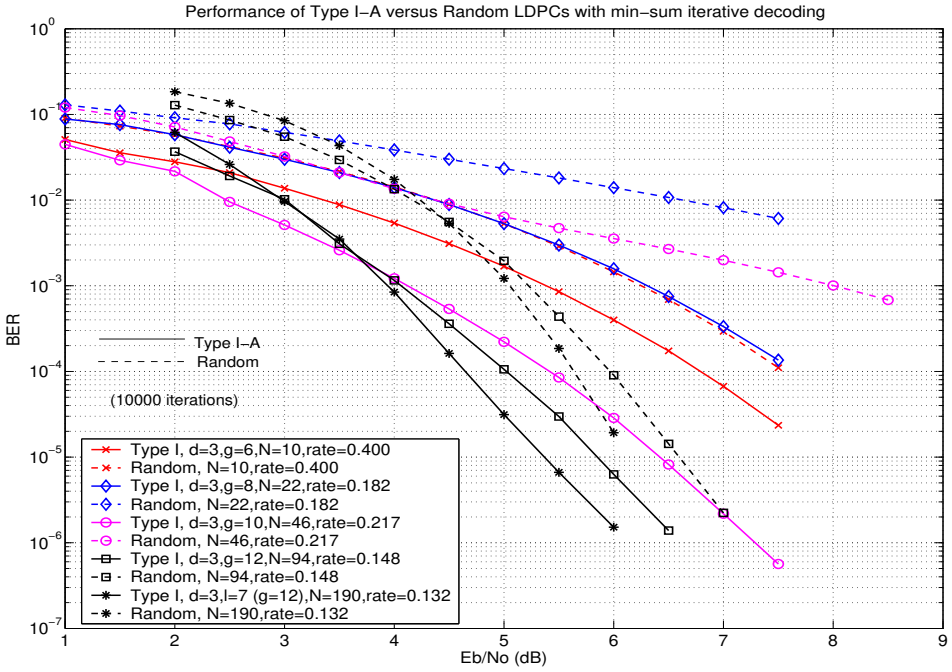


Fig. 12. Performance of Type I-A versus Random LDPC codes on the BIAWGNC with min-sum iterative decoding.

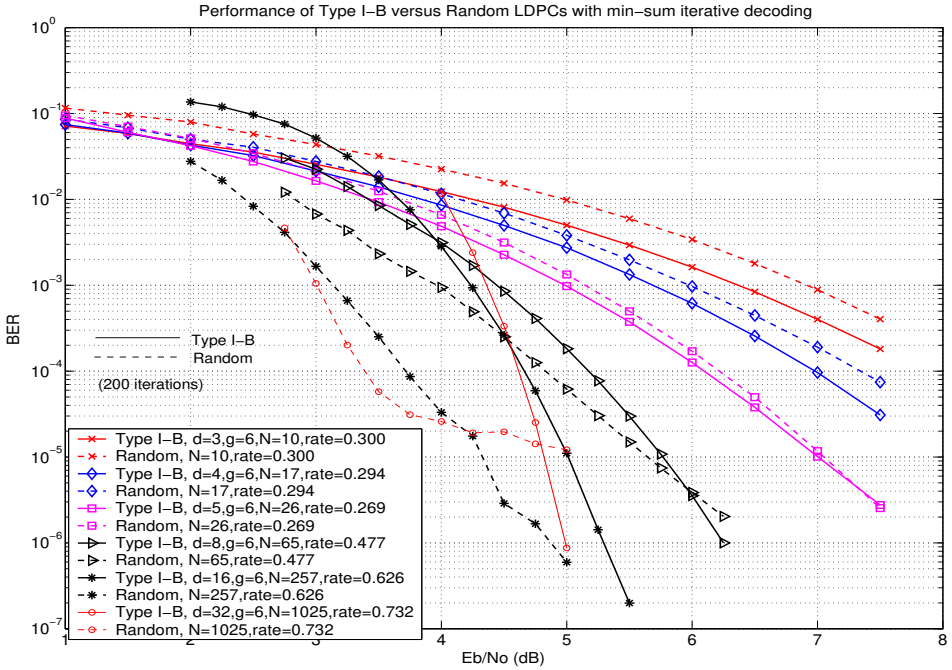


Fig. 13. Performance of Type I-B versus Random LDPC codes on the BIAWGNC with min-sum iterative decoding.

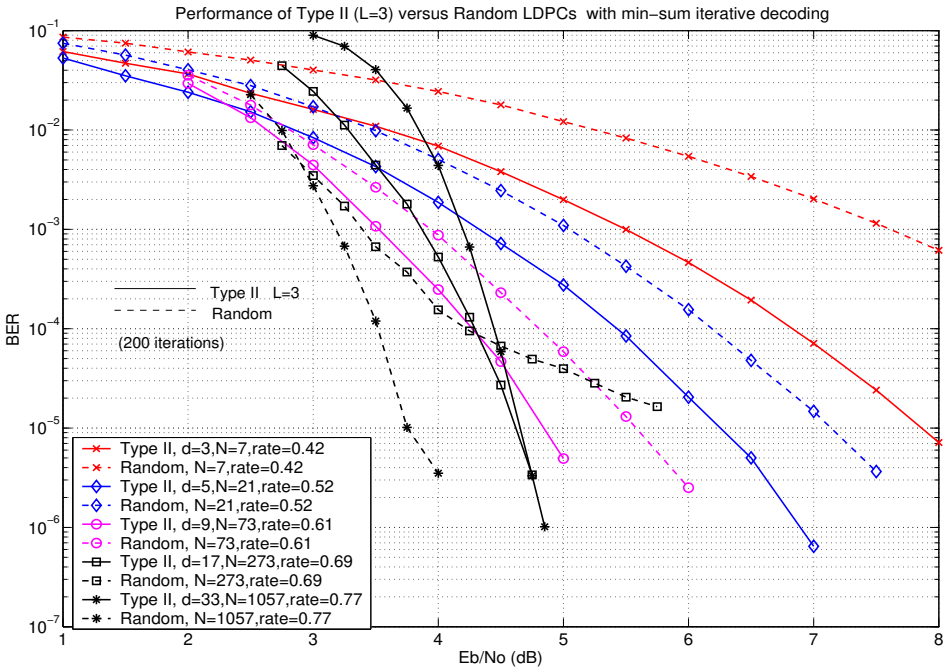


Fig. 14. Performance of Type II $\ell = 3$ versus Random LDPC codes on the BIAWGNC with min-sum iterative decoding.

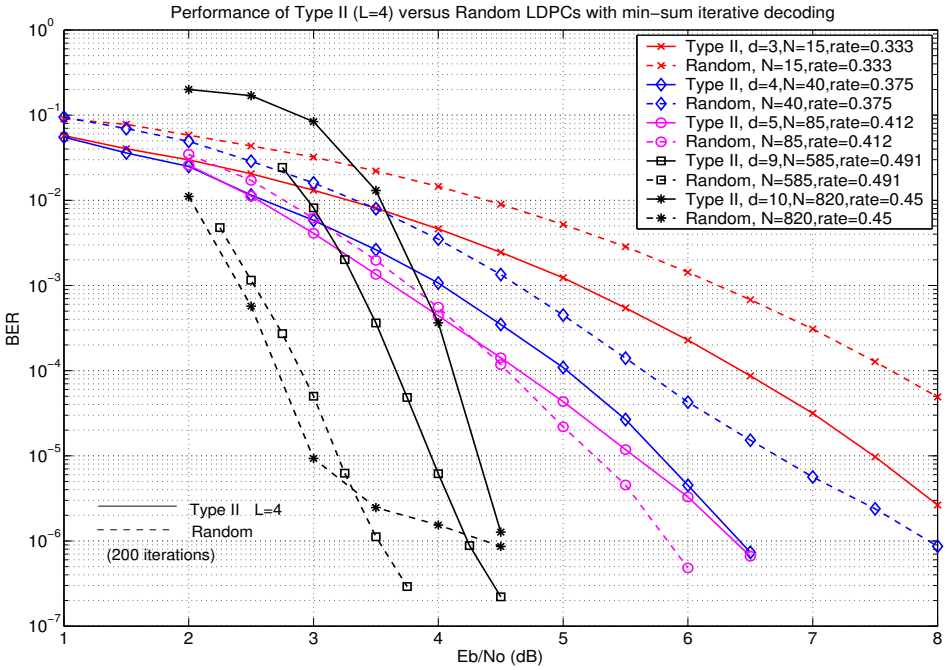


Fig. 15. Performance of Type II $\ell = 4$ versus Random LDPC codes on the BIAWGNC with min-sum iterative decoding.

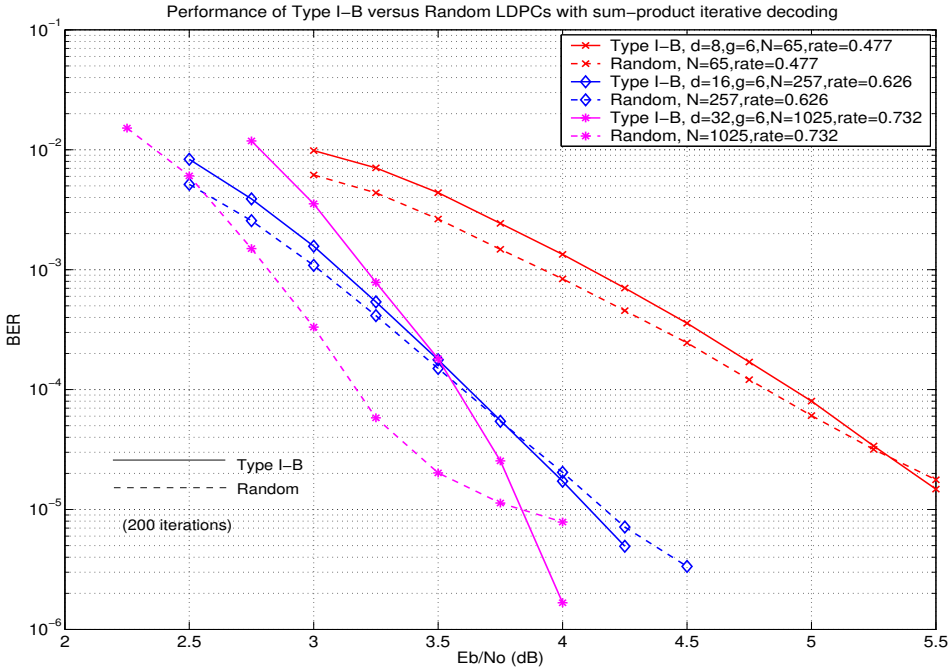


Fig. 16. Performance of Type I-B versus Random LDPC codes on the BIAWGNC with sum-product iterative decoding.

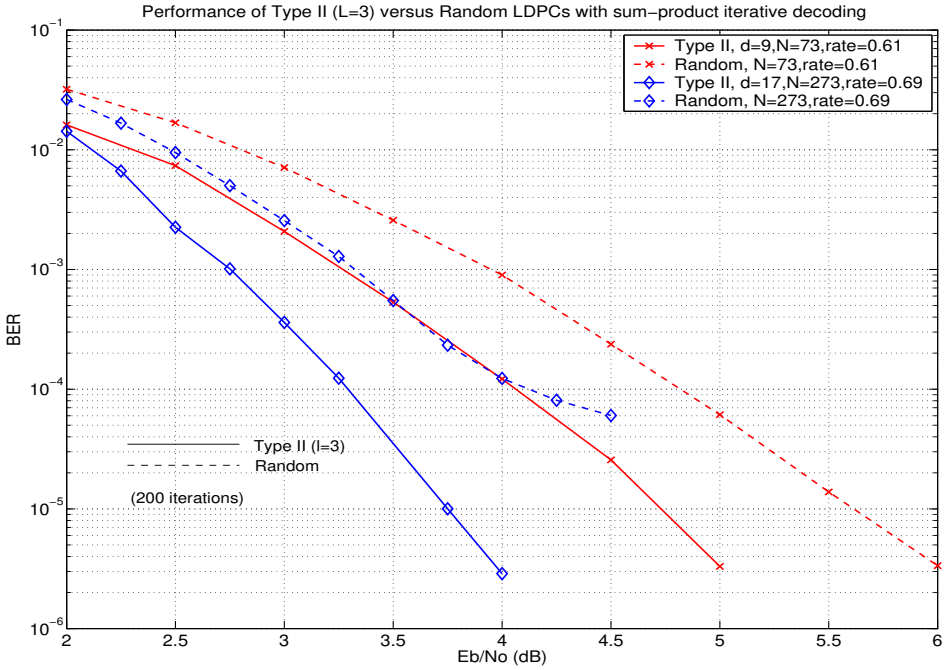


Fig. 17. Performance of Type II $\ell = 3$ versus Random LDPC codes on the BIAWGNC with sum-product iterative decoding.

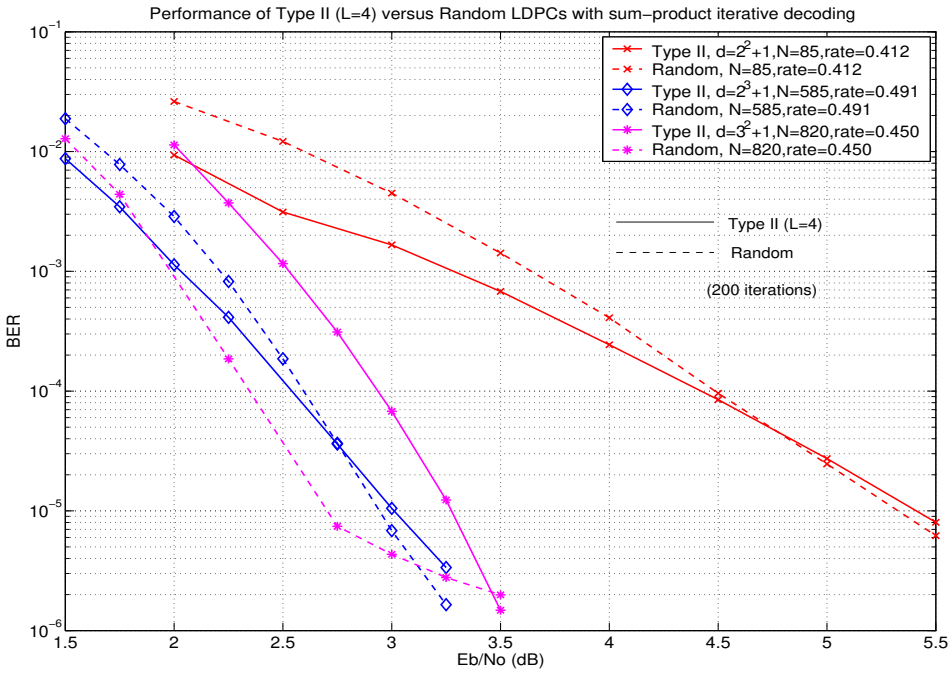


Fig. 18. Performance of Type II $\ell = 4$ versus Random LDPC codes on the BIAWGNC with sum-product iterative decoding.

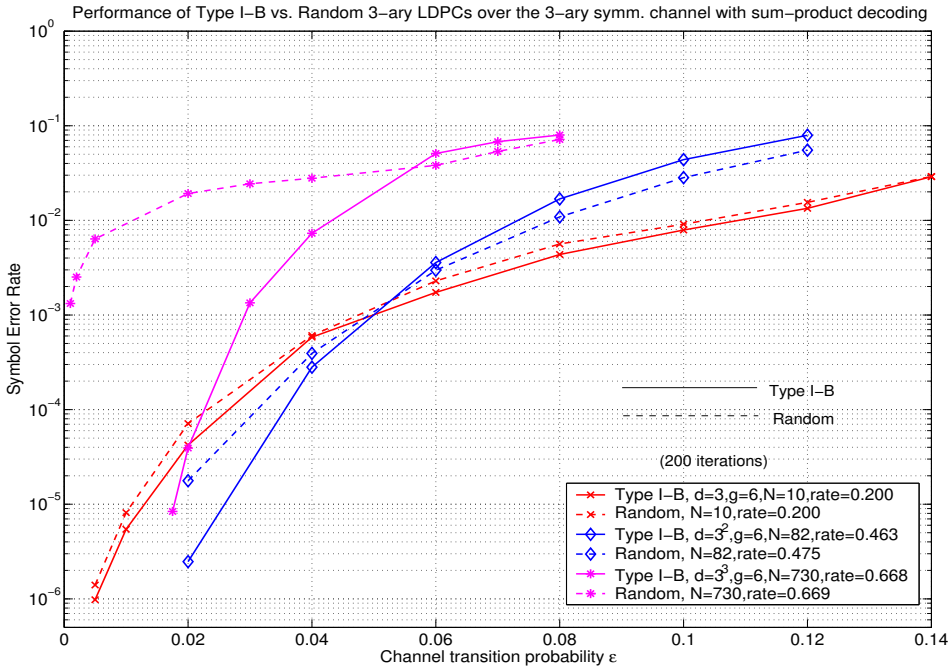


Fig. 19. Performance of Type I-B versus Random 3-ary LDPC codes on the 3-ary symmetric channel with sum-product iterative decoding.

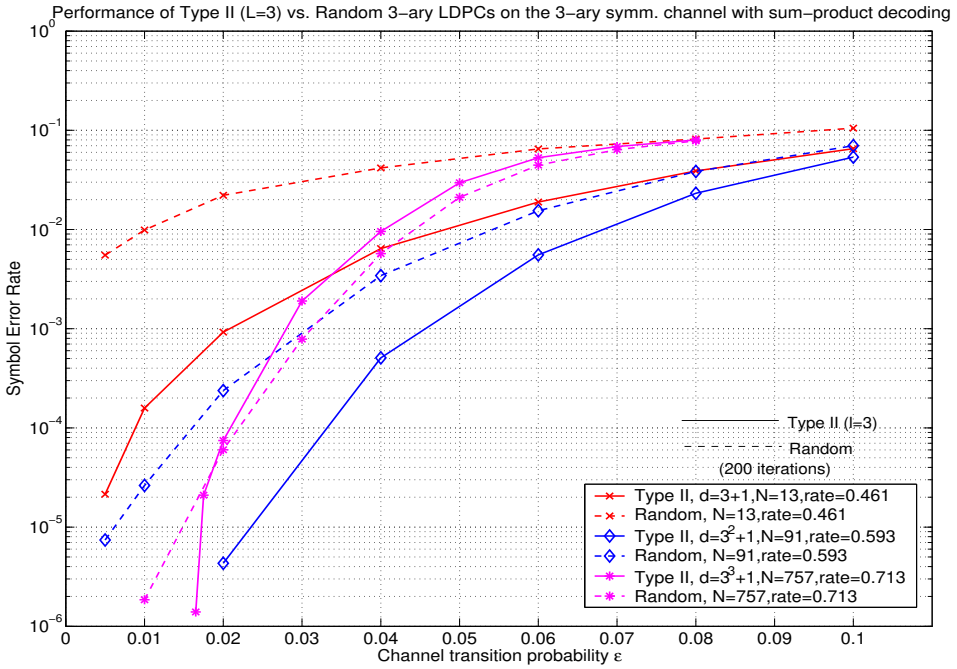


Fig. 20. Performance of Type II $\ell = 3$ versus Random 3-ary LDPC codes on the 3-ary symmetric channel with sum-product iterative decoding.

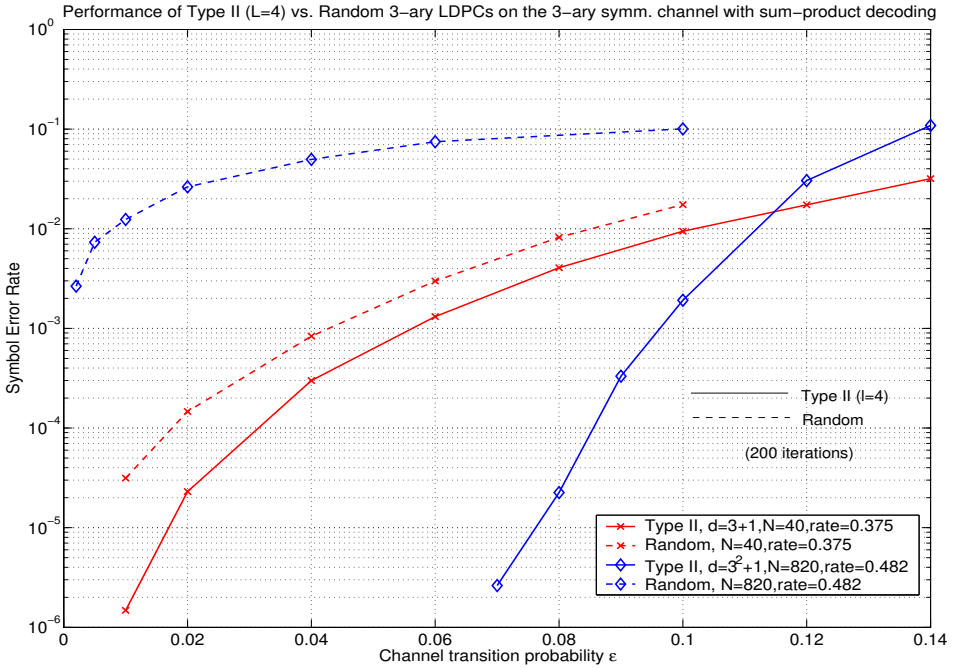
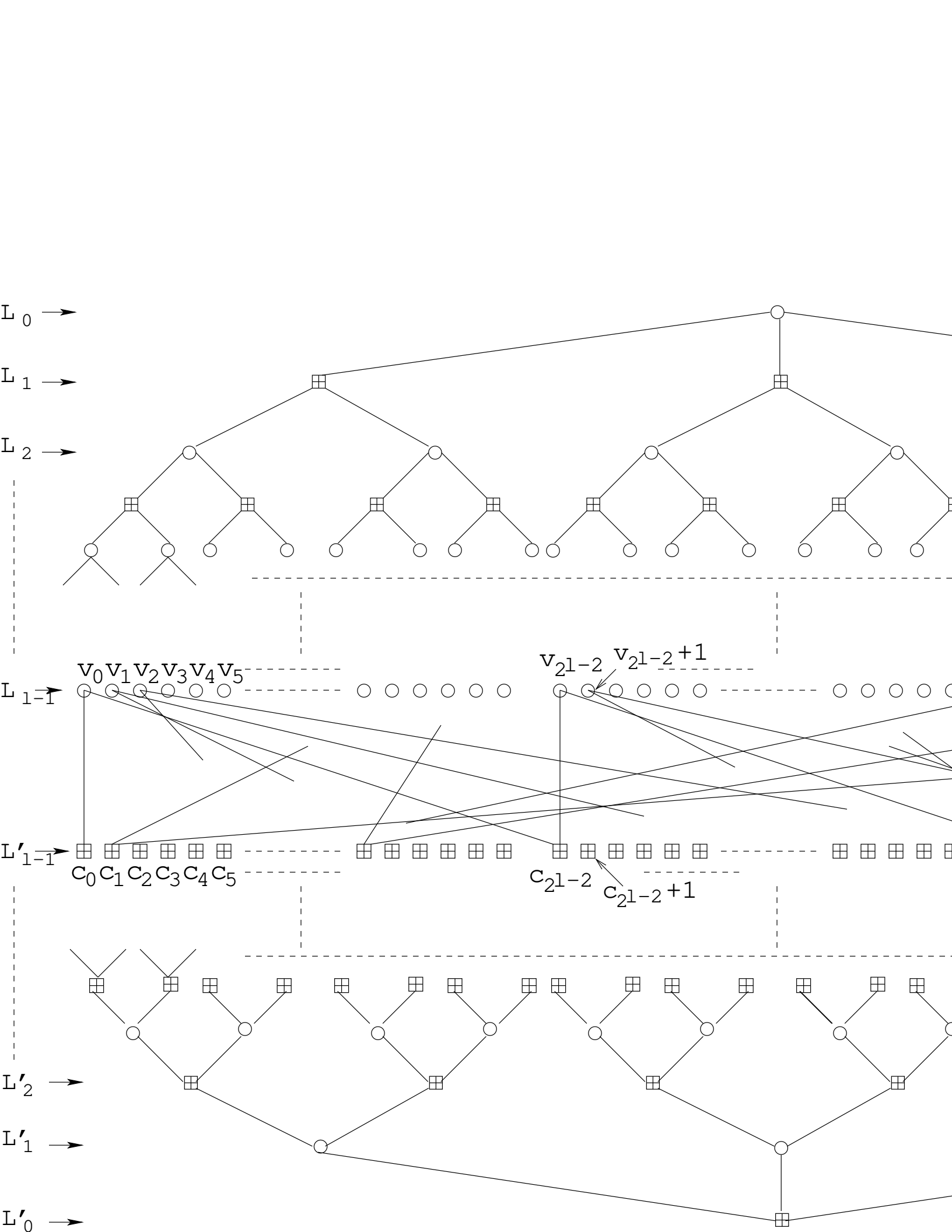
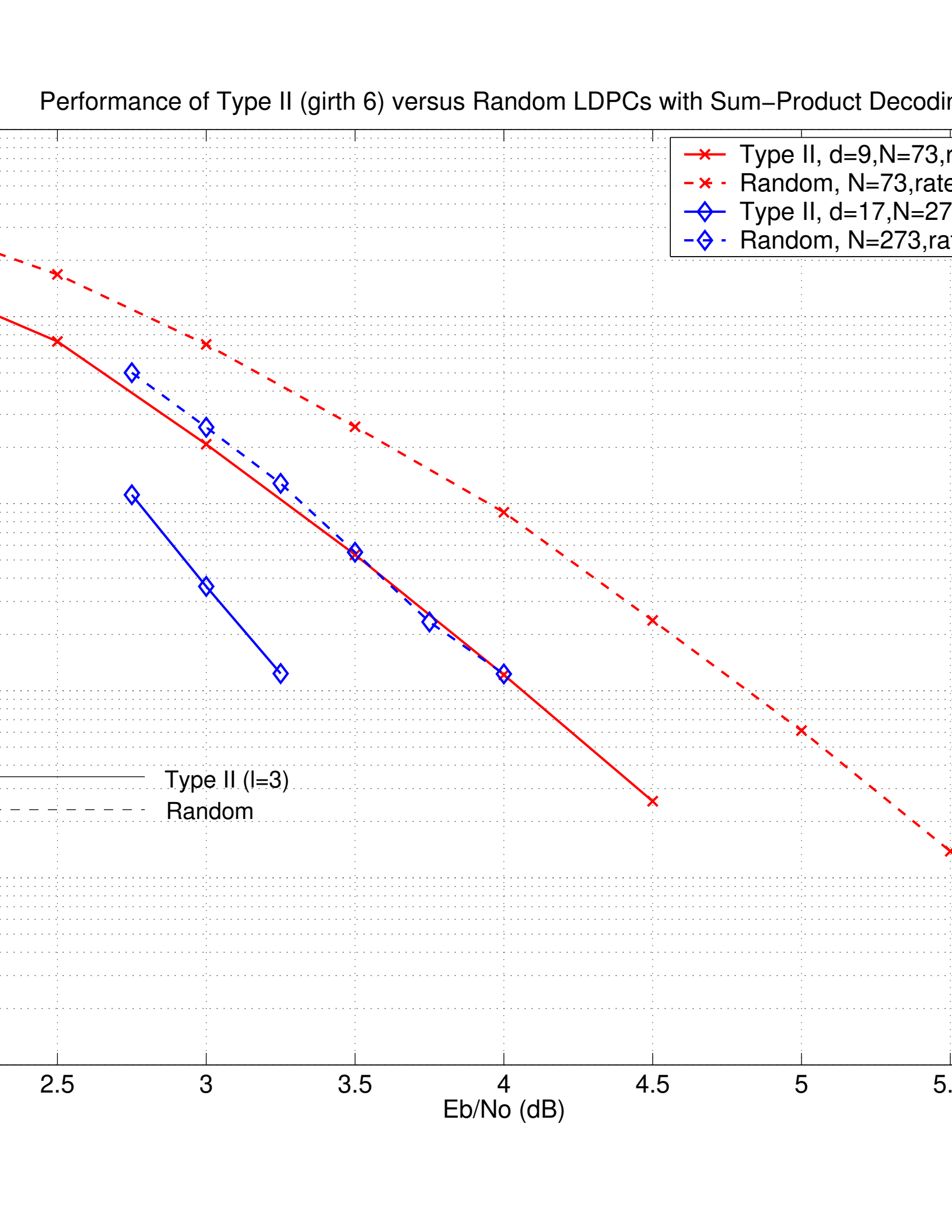
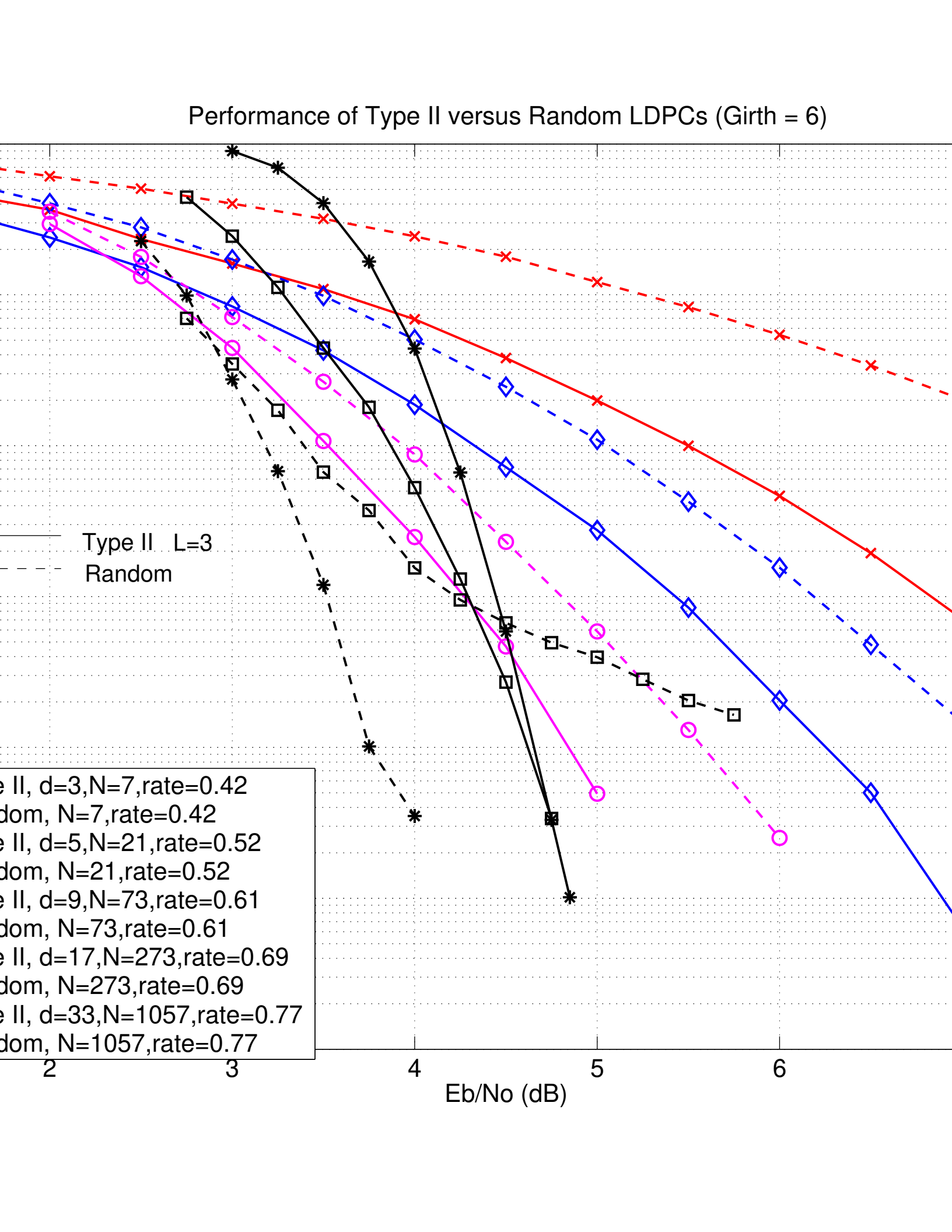
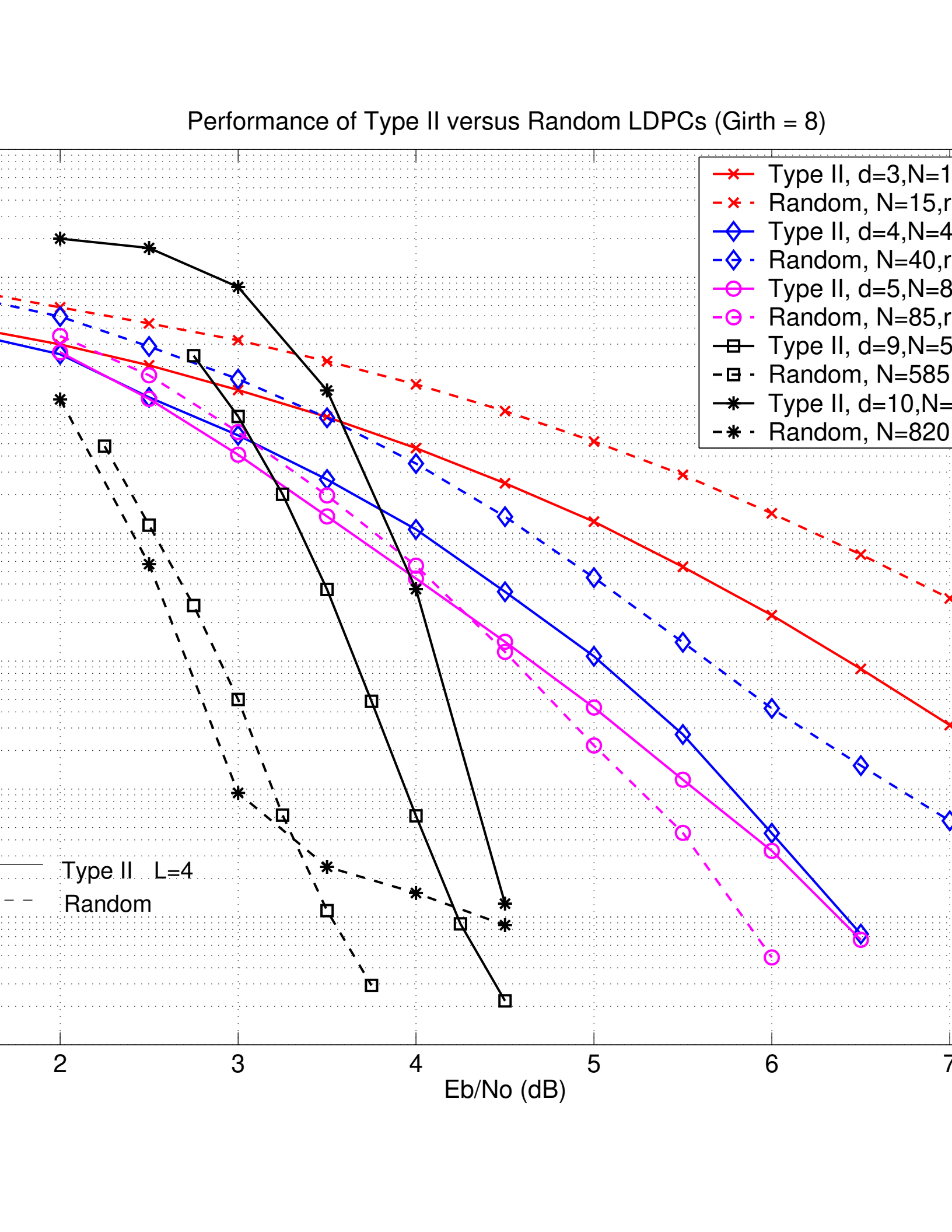


Fig. 21. Performance of Type II $\ell = 4$ versus Random 3-ary LDPC codes on the 3-ary symmetric channel with sum-product iterative decoding.


L 0
L 1
L 2
L l-1
v0 v1 v2 v3 v4 v5
v2l-2
v2l-2 +1
L'l-1
c0 c1 c2 c3 c4 c5
c2l-2
c2l-2 +1
L'2
L'1
L'0


Performance of Type II (girth 6) versus Random LDPCs with Sum−Product Decodir
Type II, d=9,N=73,r
Random, N=73,rate
Type II, d=17,N=27
Random, N=273,ra
Type II (l=3)
Random
2.5
3
3.5
4
4.5
5
5.
Eb/No (dB)


Performance of Type II versus Random LDPCs (Girth = 6)
Type II L=3
Random
e II, d=3,N=7,rate=0.42
dom, N=7,rate=0.42
e II, d=5,N=21,rate=0.52
dom, N=21,rate=0.52
e II, d=9,N=73,rate=0.61
dom, N=73,rate=0.61
e II, d=17,N=273,rate=0.69
dom, N=273,rate=0.69
e II, d=33,N=1057,rate=0.77
dom, N=1057,rate=0.77
2
3
4
5
6
Eb/No (dB)


Performance of Type II versus Random LDPCs (Girth = 8)
Type II, d=3,N=1
Random, N=15,r
Type II, d=4,N=4
Random, N=40,r
Type II, d=5,N=8
Random, N=85,r
Type II, d=9,N=5
Random, N=585
Type II, d=10,N=
Random, N=820
Type II L=4
Random
2
3
4
5
6
7
Eb/No (dB)